UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE BIOLOGIA ANIMAL

Evolving meaning: using Genetic Programming to learn similarity perspectives for mining biomedical data

Rita Isabel Torres de Sousa

Mestrado em Bioinformática e Biologia Computacional

Dissertação orientada por:
Prof.ª Doutora Cátia Luísa Santana Calisto Pesquita
Prof.ª Doutora Sara Guilherme Oliveira da Silva

2019

Acknowledgements

First, I would like to thank to my supervisors, Prof. Cátia Pesquita and Prof. Sara Silva, for guiding my work and always being available to help. Without their dedication and support this dissertation would not have been possible. I am truly grateful to them.

I would also like to express my gratitude to my grandmother, my parents and my sister for their unconditional support throughout my life. I extend my thanks to Diana, Margarida, Sofia and Teresa for their friendship. They were always there with a word of encouragement or listening ear.

Finally, I would like to thank to Fundação para a Ciência e a Tecnologia, which provides the funding under LASIGE Strategic Project (UID/CEC/00408/2019) and through the SMILAX, PERSEIDS, PREDICT, and BINDER projects (PTDC/EEI-ESS/4633/2014, PTDC/EMS-SIS/0642/2014, PTDC/CCI-CIF/29877/2017, PTDC/CCI-INF/29168/2017).

Resumo

Nos últimos anos, as ontologias biomédicas tornaram-se fundamentais para descrever o conhecimento biológico na forma de grafos de conhecimento. Consequentemente, foram propostas várias abordagens de mineração de dados que tiram partido destes grafos de conhecimento. Estas abordagens baseiam-se em representações vetoriais que podem não capturar toda a informação semântica subjacente aos grafos. Uma abordagem alternativa consiste em utilizar a semelhança semântica como representação semântica. No entanto, como as ontologias podem modelar várias perspetivas, a semelhança semântica pode ser calculada tendo em consideração diferentes aspetos. Deste modo, diferentes tarefas de aprendizagem automática podem exigir diferentes perspetivas do grafo de conhecimento. Selecionar os aspetos semânticos mais relevantes, ou a melhor combinação destes para suportar uma determinada tarefa de aprendizagem não é trivial e, normalmente, exige conhecimento especializado.

Nesta dissertação, apresentamos uma nova abordagem usando a Programação Genética sobre um conjunto de semelhanças semânticas, cada uma calculada com base num aspeto semântico dos dados, para obter a melhor combinação para uma dada tarefa de aprendizagem supervisionada. A metodologia inclui três etapas sequenciais: calcular a semelhança semântica para cada aspeto semântico; aprender a melhor combinação desses aspetos usando a Programação Genética; integrar a melhor combinação com o algoritmo de classificação.

A abordagem foi avaliada em nove conjuntos de dados para prever a interação entre proteínas. Nesta aplicação, a *Gene Ontology* foi utilizada como grafo de conhecimento para suportar o cálculo da semelhança semântica. Como referência, utilizámos uma variação da abordagem proposta com estratégias manuais frequentemente utilizadas para combinar os aspetos semânticos. Os resultados demonstraram que as combinações obtidas com a Programação Genética superaram as combinações escolhidas manualmente que emulam o conhecimento especializado. A nossa abordagem foi também capaz de aprender modelos agnósticos em relação à espécie usando diferentes combinações de espécies para treino e teste, ultrapassando assim as limitações de prever interações entre proteínas para espécies com poucas interações conhecidas.

Esta nova metodologia supera as limitações impostas pela necessidade de selecionar manualmente os aspetos semânticos que devem ser considerados para uma dada tarefa de aprendizagem. A aplicação da metodologia à previsão da interação entre proteínas foi bem-sucedida, perspetivando outras aplicações.

Palavras Chave: semelhança semântica, programação genética, ontologia, grafo de conhecimento, previsão da interacção entre proteínas.

Abstract

In recent years, biomedical ontologies have become important for describing existing biological knowledge in the form of knowledge graphs. Data mining approaches that work with knowledge graphs have been proposed, but they are based on vector representations that do not capture the full underlying semantics. An alternative is to use machine learning approaches that explore semantic similarity. However, since ontologies can model multiple perspectives, semantic similarity computations for a given learning task need to be fine-tuned to account for this. Obtaining the best combination of semantic similarity aspects for each learning task is not trivial and typically depends on expert knowledge.

In this dissertation, we developed a novel approach that applies Genetic Programming over a set of semantic similarity features, each based on a semantic aspect of the data, to obtain the best combination for a given supervised learning task. The methodology includes three sequential steps: compute the semantic similarity for each semantic aspect; learn the best combination of those aspects using Genetic Programming; integrate the best combination with a classification algorithm.

The approach was evaluated on several benchmark datasets of protein-protein interaction prediction. The quality of the classifications is evaluated using the weighted average F-measure for each dataset. As a baseline, we employed a variation of the proposed methodology that instead of using evolved combinations, uses static combinations. For protein-protein interaction prediction, Gene Ontology was used as the knowledge graph to support semantic similarity, and it outperformed manually selected combinations of semantic aspects emulating expert knowledge. Our approach was also able to learn species-agnostic models with different combinations of species for training and testing, effectively addressing the limitations of predicting protein-protein interactions for species with fewer known interactions.

This dissertation proposes a novel methodology to overcome one of the limitations in knowledge graph-based semantic similarity applications: the need to expertly select which aspects should be taken into account for a given application. The methodology is particularly important for biomedical applications where data is often complex and multi-domain. Applying this methodology to protein-protein interaction prediction proved successful, paving the way to broader applications.

Keywords: semantic similarity, genetic programming, ontology, knowledge graph, protein-protein interaction prediction.

Resumo Alargado

A descoberta de conhecimento em domínios complexos pode ser um desafio para os métodos de mineração de dados. Estes métodos são tipicamente limitados a visualizações agnósticas dos dados, não tendo acesso ao seu contexto e significado. No entanto, é amplamente reconhecido que o desempenho dos métodos de mineração de dados pode melhorar significativamente quando as relações adicionais entre os dados são tidas em conta.

Na última década, a explosão na complexidade e heterogeneidade dos dados biomédicos motivou um novo panorama de dados semânticos, onde milhões de entidades biológicas descritas semanticamente estão disponíveis em grafos de conhecimento. Os grafos de conhecimento descrevem entidades reais e as suas inter-relações, através de ligações a conceitos ontológicos que os descrevem, organizados num grafo. Cada ontologia é uma especificação formal e explícita de uma conceptualização na qual cada classe (ou conceito) está precisamente definida e as relações entre classes estão parametrizadas ou restringidas. Deste modo, as representações semânticas baseadas nos grafos de conhecimento podem ser exploradas por métodos de mineração de dados, fornecendo uma oportunidade única para melhorar os processos de descoberta de conhecimento.

Dada a crescente importância das ontologias biomédicas na forma de grafos de conhecimento, o número de abordagens que combinam métodos de mineração de dados e grafos de conhecimento tem vindo a aumentar. Um dos maiores desafios enfrentados nestas abordagens é a transformação dos dados provenientes dos grafos numa representação adequada e que possa ser processada pelos algoritmos de mineração de dados. Atualmente, destacam-se duas representações semânticas baseadas nos grafos de conhecimento designadas por *graph kernels* e *graph embeddings*. Na representação *graph kernels*, a distância entre duas instâncias depende do número de subestruturas comuns. Na representação *graph embeddings*, os grafos de conhecimento são transformados em sequências de entidades, que podem ser consideradas frases de um corpus. Posteriormente, com base no corpus, são geradas representações vetoriais usando modelos de linguagem natural. No entanto, estas representações vetoriais podem não capturar toda a informação semântica subjacente aos grafos uma vez que apenas têm em consideração subestruturas locais ou co-ocorrências. Uma abordagem alternativa consiste em utilizar a semelhança semântica como representação semântica. A semelhança semântica expressa a semelhança entre duas entidades com base no significado de cada uma. Por exemplo, se duas entidades biológicas estão anotadas com a mesma ontologia, é possível compará-las comparando as classes com as quais estão anotadas.

As ontologias visam modelar o conhecimento para um determinado domínio, mas dentro do domínio podem modelar múltiplas perspetivas, e consequentemente a semelhança semântica pode ser calculada tendo em consideração diferentes aspetos

semânticos. Deste modo, diferentes tarefas de aprendizagem automática podem exigir diferentes perspetivas do grafo de conhecimento e, consequentemente, de diferentes perspetivas da semelhança. Escolher os aspetos semânticos mais relevantes ou a melhor combinação desses aspetos para suportar uma determinada tarefa de aprendizagem não é trivial e normalmente exige conhecimento especializado.

Por exemplo, na ontologia biomédica mais usada na biologia, a *Gene Ontology*, o universo de conceitos relacionado com a função das proteínas é descrito de acordo com três aspetos diferentes: processos biológicos, componentes celulares e funções moleculares. Uma anotação consiste numa associação entre uma proteína e um conceito da *Gene Ontology*. Uma proteína pode ser anotada com vários conceitos dos três aspetos semânticos da *Gene Ontology*. Assim, é possível calcular a semelhança semântica entre duas proteínas com base nas anotações para cada aspeto, ou combinando os vários aspetos. Supondo que a tarefa de aprendizagem é a previsão de interações entre proteínas, é expectável que as semelhanças de processos biológicos e de componentes celulares sejam indicadores mais fortes de interação entre proteínas do que a semelhança de funções moleculares. Por esta razão, a escolha do especialista seria, provavelmente, uma combinação na qual a semelhança para processos biológicos e componentes celulares teria mais peso. No entanto, para outras tarefas de aprendizagem (por exemplo, previsão de genes associados a doença) a seleção dos aspetos semânticos mais relevantes pode não ser tão direta.

Nesta dissertação, apresentamos uma nova abordagem usando a Programação Genética sobre um conjunto de semelhanças semânticas, cada uma calculada com base num aspeto semântico dos dados, para obter a melhor combinação para uma dada tarefa de aprendizagem supervisionada. A Programação Genética é um algoritmo de computação evolucionária que é capaz de resolver problemas complexos através da evolução de populações de programas de computador, usando a evolução darwinista e a genética mendeliana como inspiração. Este algoritmo é um dos métodos mais adaptáveis e poderosos de aprendizagem automática dada a sua capacidade de pesquisa em grandes espaços de solução. Para além disso, ao contrário de outros métodos de aprendizagem automática, este algoritmo produz modelos legíveis.

A metodologia proposta inclui três etapas sequenciais: calcular a semelhança semântica para cada aspeto semântico; aprender a melhor combinação desses aspetos usando a Programação Genética; integrar a melhor combinação com o algoritmo de classificação. A qualidade das classificações é avaliada usando a média ponderada da *F-measure*.

A nova metodologia foi implementada e avaliada para a previsão de interação entre proteínas. Nesta aplicação biomédica, foi utilizado o grafo de conhecimento composto pela *Gene Ontology* e as anotações da *Gene Ontology* para suportar o cálculo da semelhança semântica. Na avaliação foram utilizados nove conjuntos de dados de quatro espécies diferentes, com diferentes números de elementos. A primeira etapa da implementação da metodologia envolve o cálculo da semelhança semântica para cada aspeto semântico da *Gene Ontology*, utilizando diferentes medidas de semelhança semântica projetadas para esta ontologia. No final desta etapa, cada instância do conjunto de dados que representa um par de proteínas fica caracterizada por três

valores correspondentes à semelhança semântica para cada um dos aspetos semânticos da *Gene Ontology*, e uma "etiqueta" (interação ou não interação). Na segunda etapa, o algoritmo de Programação Genética é usado para aprender a melhor combinação dos aspetos semânticos da *Gene Ontology*. A combinação selecionada no final da evolução é utilizada na classificação no conjunto de teste, obtendo-se um valor de desempenho. Como referência, foi utilizada uma variação da abordagem proposta com estratégias manuais frequentemente utilizadas para combinar os aspetos semânticos.

Os resultados demonstraram que, para conjuntos de dados suficientemente grandes, as combinações obtidas com a Programação Genética superam as combinações escolhidas manualmente que emulam o conhecimento especializado. Para ultrapassar a limitação do número de elementos dos conjuntos de dados, foram realizadas várias experiências com combinações de conjuntos de dados da mesma espécie. Estas experiências revelaram que utilizar mais dados, mesmo que pertencentes a outro conjunto, pode ser benéfico, no entanto foi também confirmado que cada conjunto de dados tem um enviesamento inerente. A nossa abordagem foi também capaz de aprender modelos agnósticos em relação à espécie usando diferentes combinações de espécies para treino e teste, ultrapassando assim as limitações de prever interações entre proteínas para espécies com poucas interações conhecidas. Quanto à análise dos modelos obtidos para cada conjunto de dados, os resultados mostraram estar em concordância com resultados obtidos com outros métodos de previsão.

Esta nova metodologia supera uma das limitações das aplicações baseadas na semelhança semântica em grafos de conhecimento: a necessidade de selecionar manualmente os aspetos que devem ser considerados para uma dada aplicação. Esta metodologia é particularmente importante para aplicações biomédicas em que os dados são geralmente complexos. A aplicação da metodologia à previsão da interação entre proteínas foi bem-sucedida, perspetivando outras aplicações biomédicas (por exemplo, descoberta de genes associados a doenças). Como trabalho futuro, pretendemos adicionar mais medidas de semelhança semântica à avaliação, aplicar a metodologia a outras tarefas de aprendizagem e combinar a abordagem proposta para selecionar os aspetos semânticos mais relevantes usando outras abordagens baseadas, por exemplo, em *graph embeddings*.

Contents

List of Figures

List of Tables

Acronyms

AA All Ancestors.

AUC-ROC Area Under the Receiver Operating Characteristic Curve.

BP Biological Process.

CC Cellular Component.

DAG Directed Acyclic Graph.

GAF Gene Association File.

GO Gene Ontology.

GOA Gene Ontology Annotation.

GP Genetic Programming.

HPO Human Phenotype Ontology.

IC Information Content.

KDD Knowledge Discovery in Databases.

KG Knowledge Graph.

LOD Linked Open Data.

MF Molecular Function.

OBO Open Biomedical Ontology.

OWL Web Ontology Language.

PICR Protein Identifier Cross-reference.

PPI Protein-Protein Interaction.

RDF Resource Description Framework.

REST Representational State Transfer.

RMSE Root Mean Square Error.

SML Semantic Measures Library.

SS Semantic Similarity.

SSM Semantic Similarity Measure.

SVM Support Vector Machine.

ULCA Up to Lowest Common Ancestor.

URL Uniform Resource Locator.

WAA Weighted All Ancestors.

WAF Weighted Average F-measure.

WULCA Weighted Up to Lowest Common Ancestor.

Chapter 1

Introduction

The amount and complexity of biological data that is being collected and accumulated is increasing at accelerated rates due to improvements of existing technologies and the introduction of new ones. Given the exponential growth of biological data, there is an urgent need for a new generation of computational tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of data. These tools are the subject of the research field of knowledge discovery in databases (KDD), which aims at transforming low-level data into other forms that might be more abstract or more useful (for example, to obtain a predictive model) (Fayyad *et al.*, 1996). The tasks performed in this research field are knowledge-intensive and can often benefit from using additional knowledge from various sources.

Data mining is a particular step in the process of discovering useful knowledge from data and is defined as a non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. The idea of data mining is to build computer programs that go through databases automatically, looking for patterns or regularities. Strong patterns, if found, can be generalized to make accurate predictions on future data. However, many patterns will be uninteresting or accidental coincidences in the particular used dataset (Fayyad *et al.*, 1996). Data mining uses algorithms and techniques from statistics, machine learning, databases and data warehousing, and many other disciplines to analyze large datasets. Classification, clustering, and regression are the most popular tasks in data mining. The choice of the algorithm and technique for each task depends on the nature of the data as well as the desired knowledge (Tzanis *et al.*, 2008).

In bioinformatics, the discovery of new biologically relevant patterns depends on the comparison and integration of massive datasets that often contain complex data. Knowledge discovery in complex domains can be a challenge for data mining methods, which are typically limited to agnostic views of the data, without being able to gain access to its context and meaning. It is widely recognized that the performance of data mining methods can improve significantly when additional relations among the data objects are taken into account, a strategy employed in relational data mining and Inductive Logic Programming (De Raedt, 2008).

In the last decade, the explosion in complexity and heterogeneity of biological data has motivated a new panorama of semantic data, where millions of semantically-described biological

entities are available in knowledge graphs (KGs) (Schmachtenberg *et al.*, 2014). KGs describe real-world entities and their interrelations, through links to ontology concepts describing them, organized in a graph (Ehrlinger & Wöß, 2016). The Linking Open Data cloud diagram for Life Sciences domain[1] (Figure 1.1) illustrates the vast number of datasets published in linked data format that is available. Therefore, semantic representations of data entities based on KGs that can be explored by data mining approaches provide a unique opportunity to enhance knowledge discovery processes.

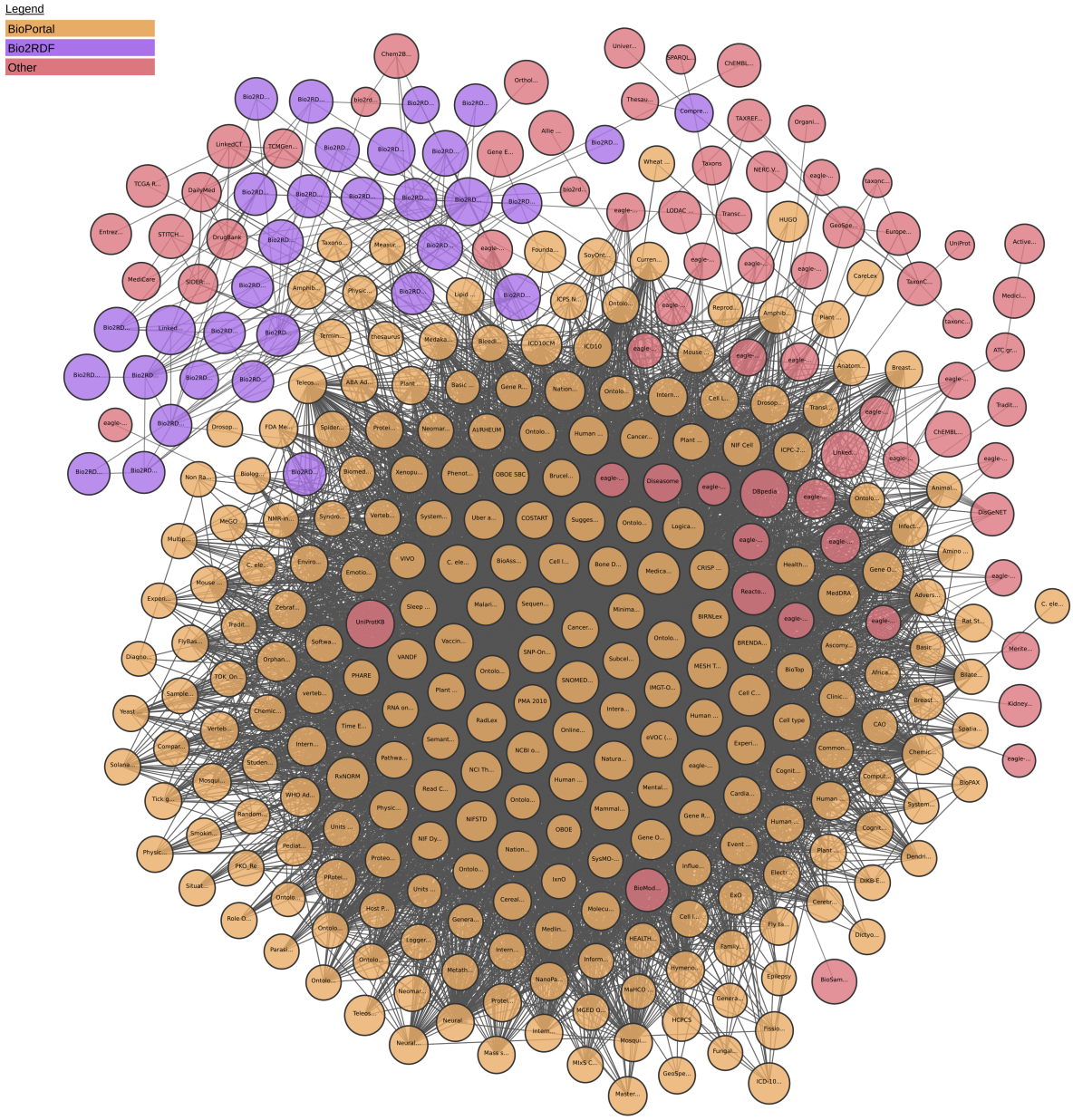

Figure 1.1: LOD subcloud of Life Sciences domain.

[1] https://lod-cloud.net/

One of the biggest challenges faced by the approaches that combine methods from data mining and knowledge discovery with KGs is how to transform data coming from KGs into a suitable representation that can be processed by those methods. Most of the existing approaches build a propositional feature vector representation of the data (i.e., each instance is represented as a vector of features), which allows the subsequent application of most existent data mining algorithms. However, the approaches based on vector representations may fail to capture the full underlying semantics. For instance, the state-of-art approaches (graph embeddings and graph kernels) mostly explore the local structure of KGs and co-occurrences. An alternative strategy, and since measuring similarity is fundamental to many machine learning algorithms, is to use the KGs to measure the semantic similarity (SS) between entities in the graph. SS is the computation of the similarity between entities based on their meaning as described in an ontology. For instance, if two biological entities are annotated within the same ontology, we can compare them by comparing the classes with which they are annotated (Pesquita *et al.*, 2009).

However, a challenge remains. Ontologies aim at modeling a given domain, but within a single domain there can be multiple perspectives, and the SS can be computed taking different aspects into consideration. Taking as an example the Gene Ontology (GO): it describes protein function according to three different perspectives or aspects: biological process (BP), cellular component (CC) and molecular function (MF). Therefore, we can compute the SS between two proteins in terms of their annotations within a single aspect, or combining multiple aspects. Different learning tasks may need different perspectives of the KG, and selecting the best aspects or combination of aspects to support a given learning task is not trivial. Usually, the selection of the combination of SS aspects is based on a researchers' intuition and experience. For instance, if the learning task is the prediction of interaction between proteins, it is expected that similarity in biological process or cellular component are stronger indicators for protein interaction than similarity in molecular function. Therefore, a combination in which biological process and cellular component aspects have more weight will probably be the choice of researchers. However, not all tasks have such a clear choice of combination. For instance, if the learning task is the prediction of disease-associated genes, how to combine molecular function with the remaining two aspects is not straightforward.

1.1 Objectives

Adjusting the combination of SS aspects to the machine learning task represents a challenge. A serious limitation of existing approaches for machine learning using SS as KG-based representation is that the choice of the suitable combination of the SS aspects for a given learning task depends on manual selection. Automating the selection of the best combination of KG aspects to support specific tasks would simplify and generalize the application of these techniques, rendering it more independent of expert knowledge.

The main goal of this dissertation is to propose a novel methodology that uses Genetic Programming (GP) (Poli *et al.*, 2008) over a set of semantic similarities, each computed over a different semantic aspect of the underlying data, to arrive at the best combination between the different aspects to support different supervised learning tasks. The underlying hypothesis is that GP can learn suitable combinations of SS aspects to support specific learning tasks. GP was

chosen for its unmatched ability to search large solution spaces by means of evolving a population of free-form readable models through crossover and mutation.

This methodology was applied to Protein-Protein Interaction (PPI) prediction, where the relationships between the different semantic aspects and potential classification performance are well established. Furthermore, the use of similarity between two proteins to predict whether they interact is one of the most straightforward ways of basing classification problems on SS values.

1.2 Contributions

The main contributions of this dissertation are:

1. Development of a novel GP-based approach to learn a suitable combination of SS aspects for specific machine learning applications;
2. Implementation of the novel approach for PPI prediction;
3. Comparative evaluation of existing approaches combining KGs with machine learning;
4. Poster with the preliminary results presented in 4th LASIGE Workshop, which was awarded the Best Student Poster Award;
5. Oral presentation of the main results at the 53rd Annual Scientific Meeting of the European Society for Clinical Investigation, in Coimbra. The abstract of this presentation was also published in Book of Abstracts of European Journal of Clinical Investigation;
6. Submission of a scientific article titled "Evolving knowledge graph similarity for supervised learning in complex biomedical domains" for the special issue on Machine Learning and Artificial Intelligence in Bioinformatics of BMC Bioinformatics.

1.3 Document Structure

The present introductory chapter gives a contextualization of the problem underlying the proposed hypothesis and introduces the main objectives and contributions of this dissertation. The remaining five chapters are organized as follows. Chapter 2 defines and explains the foundational concepts needed to understand the problem itself. Chapter 3 surveys the relevant work developed in this field to this date. Chapter 4 presents an overview of the proposed methodology with a description of the main tasks. Chapter 5 presents one biomedical application of the proposed methodology, including resources used, methods, results, and discussion. Chapter 6 summarizes the main conclusions of this work, debating some of the limitations and how to address them in the future.

Chapter 2

Concepts

For the sake of completeness, this chapter introduces a set of concepts required to understand the presented work on the topics of semantic web and genetic programming.

2.1 Semantic Web

The term Semantic Web was introduced in 2001 by Tim Berners-Lee to mean "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee *et al.*, 2001). For the semantic web to function, computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning.

Since the beginning, the Semantic Web has promoted a graph-based representation of knowledge. Such KGs contain factual knowledge about real-world entities and the relations between them (linked data) in a fully machine-readable format. Ontologies can be used to describe the entities in the KGs, providing the appropriate support to measure the SS between them. The next sections present an overview of the basis of linked data, ontologies, KGs and SS.

2.1.1 Linked Data

The term linked data can be viewed as a subset of the semantic web concept and refers to a set of best practices for publishing and connecting structured data on the Web (Bizer *et al.*, 2011). The adoption of the linked data best practices led to a global data space connecting data from diverse domains such as proteins, genes, drugs, scientific publications, people, companies, books, films, music, television and radio programs.

Resource description framework (RDF) is a common data model for linked data that defines how to express relationships between arbitrary data elements. In RDF terminology, a statement is a small piece of knowledge in the format of subject-predicate-object expressions. Figure 2.1 depicts the structure of a RDF statement. These expressions are known as triples in RDF terminology. Subject and object are two things and predicate is the name of a relation that

connects these two things. Predicate relates the subject to another entity or provides information about the entity itself.

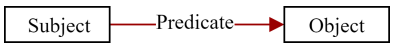

Figure 2.1: Structure of a RDF statement.

The main example of the use of linked data is the Linked Open Data (LOD) community project. So far it has resulted in an openly interlinked collection of datasets in machine-interpretable form, comprising data from diverse domains, including geography, media, and life sciences. As shown in the LOD cloud diagram, life science data occupies one of the major domains of LOD (see Figure 1.1). This situation was primarily brought by the Bio2RDF project (Belleau *et al.*, 2008) which translated major public bioinformatics databases into RDF.

2.1.2 Ontologies and Semantic Annotation

As a means to express knowledge about a domain in the Semantic Web, ontologies have been introduced in the early 1990s. In the computer science context, an ontology is an explicit specification of a conceptualization in which each element is precisely defined and the relationships between elements are parameterized or constrained (Schmachtenberg *et al.*, 2014). Ontologies are thus semantic models for reality domains.

The two components of ontologies are: (i) a set of concepts (or classes) that define the entities in a domain; and (ii) a set of semantic links between the classes that describe interactions between classes or properties of classes. Ontologies often structure their classes and the relationships between them as a directed acyclic graph (DAG), where the classes are nodes and relationships are edges.

Since ontologies are abstractions over reality, they only contain facts that are true for all entities of a particular type. For that reason, they do not contain entities but instead, represent classes only. A semantic annotation is about assigning real-world entities in a domain to their semantic description (Kiryakov *et al.*, 2004). Relying on ontology classes to annotate biomedical entities allows automatic reasoning to be applied directly to them. For instance, proteins are annotated with their functions using GO, a very successful biomedical ontology that describes the universe of concepts related to gene function, as we can see in Figure 2.2. These annotations can be seen as a semantic description of the protein, since they can be used to, computationally, assign to the protein a meaning.

One important aspect of computational ontologies is the notion that an ontology provides semantics (i.e., meaning) to the entities it represents. However, the meaning is not described explicitly, as happens for example in dictionaries, but rather is described in the relationships between the classes and in the overall structure of the ontology. Consequently, the power of ontologies lies in their capacity to capture knowledge about a domain in a shareable and computationally accessible form (Schuurman & Leszczynski, 2008).

The life sciences field has been taking advantage of ontologies for the past decades. Not only is the number of ontologies increasing, their size is also growing, their relevance in biomedical research is rising and they penetrate more areas of biology and biomedicine. Biomedical ontologies

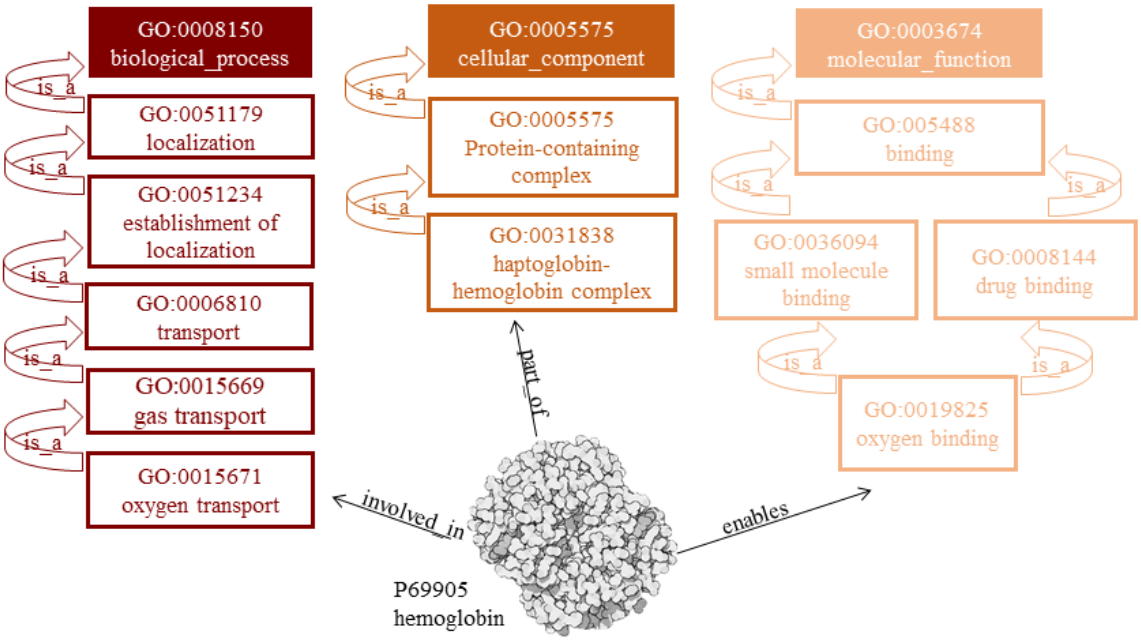

Figure 2.2: Graph representation of part of GO and GO Annotations.

have an essential role in the representation of knowledge in a computer-comprehensible way, interoperability across databases and data integration. These ontologies are used in areas ranging from gene function, as seen in the GO, to those used in characterization of drugs (Degtyarenko *et al.*, 2007). Phenotype ontologies (Robinson *et al.*, 2008) are also available for multiple species and are widely used for the annotation of the abnormalities observed in mutagenesis experiments as well as for the characterization of diseases. Open repositories such as the BioPortal (Whetzel *et al.*, 2011) provide access to hundreds of biomedical ontologies expressed in various formats, e.g., RDF, Open Biomedical Ontology (OBO), Web Ontology Language (OWL).

2.1.3 Knowledge Graphs

Ontologies and linked data are usually represented by graphs which are designated KGs. These graphs provide a conceptualization of a domain based on a formal definition of its entities and their relations.

In other words, KGs describe real-world entities and their interrelations, through links to ontology classes describing them, organized in a graph (Ehrlinger & Wöß, 2016). The nodes of KG are employed in representing ontology classes and RDF statements' subjects and objects, and edges are employed in representing ontology classes' relations and RDF statements' predicates. For example, GO and its associated annotations that link proteins to GO classes and to other proteins make up a KG. Figure 2.3 shows a small example graph of that KG.

KGs represent an unparalleled opportunity for machine learning, given their ability to provide meaningful context to the data through semantic representations. KGs provide multiple

perspectives over an entity, describing it using different properties or multiple portions of the graph.

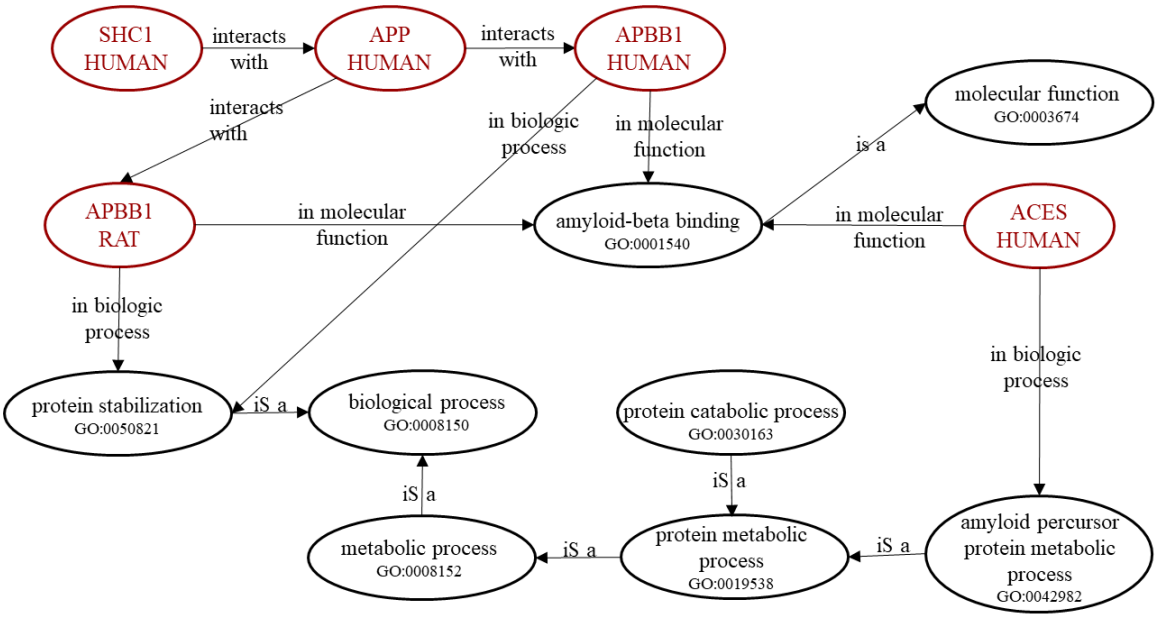

Figure 2.3: Subgraph of the GO KG illustrating the relationships between proteins. The red nodes are the biological entities (proteins) and the black nodes are the ontology concepts (GO classes).

2.1.4 Semantic Similarity

A semantic similarity measure (SSM) is a function that, given two ontology classes or two sets of classes annotating two entities, returns a numerical value reflecting the closeness in meaning between them (Pesquita *et al.*, 2009). The meaning of the classes being compared is automatically extracted from the ontologies. In the case of GO and GO annotations, SS can be calculated for two ontology classes, for instance calculating the similarity between two GO classes (e.g., the GO term *protein metabolic process* and the GO term *protein stabilization*); or between two entities each annotated with a set of classes, for instance calculating the similarity between two proteins. Each protein can be annotated with several GO terms within each of the three GO aspects so, to assess the SS between proteins (within a particular GO aspect) it is necessary to compare sets of terms rather than single terms. Figure 2.4 illustrates how two proteins are represented by their GO terms when some terms annotate only one protein while others annotate both proteins.

The approaches used to quantify SS can be distinguished based on which entities they intend to compare: approaches for comparing two terms and approaches for comparing two entities each annotated with a set of terms.

There are essentially two types of measures for comparing terms in a graph-structured ontology:

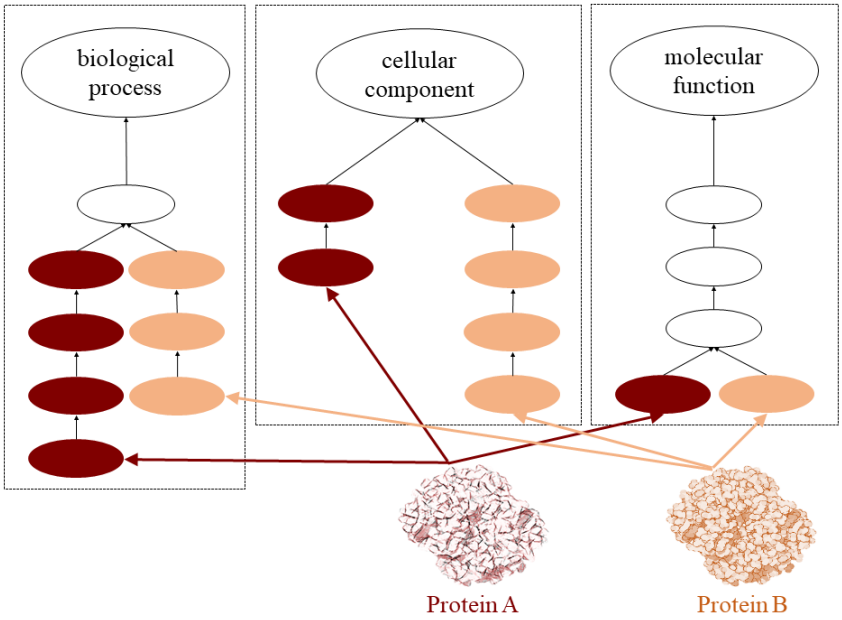

Figure 2.4: Illustration of a DAG representing GO terms annotating two proteins. Red terms annotate only protein A, orange terms annotate only protein B and white terms annotate both proteins A and B.

- Edge-based measures consist mainly on counting the number of edges in the graph path between two terms. The most common technique, distance, selects either the shortest path or the average of all paths when more than one path exists (Pesquita *et al.*, 2009). Most of these measures assume that the distance between all the relationships in an ontology is constant or depth-dependent. Neither assumption is valid in existing biomedical ontologies so edge-based measures are rarely used in the biomedical field.
- Node-based measures depend on comparing the properties of the terms involved, which can be related to the terms themselves, their ancestors, or their descendants. More recent measures explore the notion of information content (IC), a measure of how specific and informative a class is. This gives SSMs the ability to weight the similarity of two classes according to their specificity. IC can be calculated based on intrinsic properties, such as the structure of the ontology, or using external data, such as the frequency of annotations of entities in a corpus. Taking Figure 2.3 as an example, this allows SSMs to consider *protein catabolic process* and *amyloid precursor protein metabolic process* more similar than *protein metabolic process* and *protein stabilization*.

Calculating SS for two entities each annotated with a set of classes typically employs one of two approaches:

- Pairwise – where pairwise comparisons between all classes annotating each entity are considered;
- Groupwise – where set, vector or graph-based measures are employed, circumventing the need for pairwise comparisons.

Many SSMs applied to biomedical ontologies, especially to GO, have been proposed, see for instance Guzzi *et al.* (2011); Harispe *et al.* (2014); Pesquita (2017) and references therein.

2.2 Genetic Programming

GP (Poli *et al.*, 2008) is the master algorithm of evolutionary computation (Domingos, 2015) and one of the most adaptable, powerful, underused and misunderstood methods of machine learning. GP is capable of solving complex problems by evolving populations of computer programs, using Darwinian evolution and Mendelian genetics as inspiration.

This evolutionary computation technique, given all the elements of a programming language, has the potential to find the computer program that solves a particular problem without requiring the user to know or specify the form or structure of the solution in advance. Theoretically, GP can solve any problem whose candidate solutions can be measured and compared. It normally evolves solutions that are competitive with the ones developed by humans (Koza, 2010), and sometimes surprisingly creative. GP implicitly performs automatic feature selection, as selection promptly discards the unfit individuals, keeping only the ones that supposedly contain the features that warrant a good fitness. Unlike other powerful machine learning methods (e.g., Deep Learning), GP produces readable 'white-box' models.

Figure 2.5 illustrates the basic GP evolutionary cycle. Starting from an initial population of randomly created programs/models representing the potential solutions to a given problem, it evaluates and attributes a fitness value to each of them, quantifying how well the program/model solves the problem. Fitness can be measured in many ways. For example, in terms of the amount of error between its output and the desired output or in terms of the F-measure of the program in classifying objects. New generations of programs are iteratively created by selecting parents based on their fitness and breeding them using (independently applied) genetic operators like crossover (swapping of randomly chosen parts between two parents, thus creating two offspring) and mutation (modification of a randomly chosen part of a parent, thus creating one offspring). The fitter individuals are selected more often to breed and thus pass their characteristics to their offspring, so the population tends to improve in quality along successive generations. This evolutionary process continues until a given stop condition is verified (e.g., maximum number of generations, or fitness reaching some threshold), after which the individual with the best fitness is returned as the best model found.

In tree-based GP (the most common type), models are represented as parse trees that are readily translated to readable strings. For example Figure 2.6 shows the syntax tree representing $max(X0 + X0, X0 + 3 \times X1)$. The variables and constants $(X0, X1, 3)$ in the model constitute what is called the terminal set in GP, as they are only admitted as terminal nodes of the trees. In contrast, the function set contains the arithmetic operators $(max, +, \times)$ that can be used to combine elements (terminals and functions), and can only appear in internal nodes of the trees. The function set is a crucial element in GP. Together with the fitness function and the genetic operators, it determines the size and shape of the search space.

Given the free-form nature of the models evolved by GP, its intrinsic stochasticity, and the size of the search space where it normally operates, there is high variability among the raw

models returned in different runs, even when using the same settings and same dataset. Even upon simplification, these models normally remain structurally very different from each other, while possibly exhibiting similar behavior, i.e., returning similar predictions. This characteristic raises some difficulty in interpreting the GP models, even if they are fully readable. Either way, it is always advisable to run GP more than once for the same problem, to avoid the risk of adopting a sub-optimal model that may have resulted from a less successful search on such a large space.

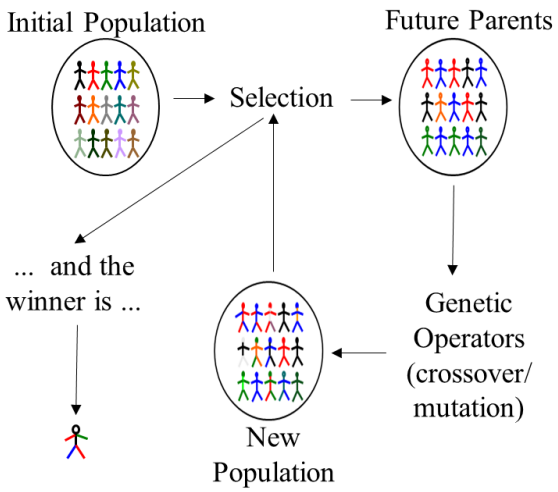

Figure 2.5: GP flowchart.

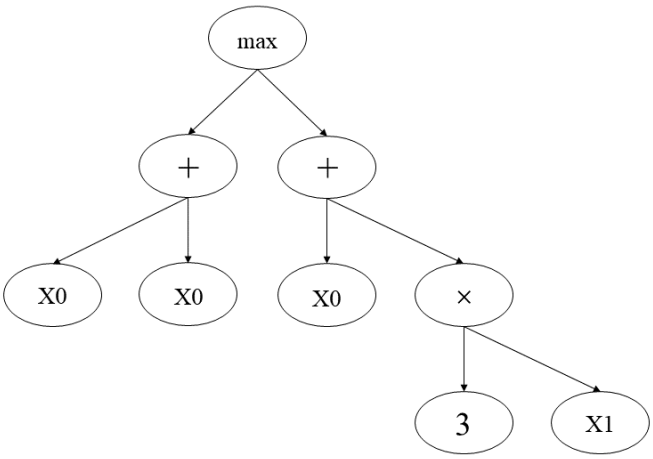

Figure 2.6: GP syntax tree example.

Chapter 3

Related Work

In this dissertation, we take advantage of the vast amount of KGs in the semantic web to enhance the KDD processes. However, graphs are not readily adaptable to common data mining tools. KG-based semantic representations bridge the gap between KGs and the typical vector-based representations of entities used by most machine learning techniques. Once a suitable representation is achieved, different machine learning algorithms can be employed.

The following sections present the state-of-art KG-based representations (graph kernels and graph embeddings) and other semantic representations based on KGs.

3.1 Graph Kernels

Given two input objects u and v, the basic idea behind kernel methods is to construct a kernel $k(u, v)$ which measures the similarity between u and v. This kernel can also be viewed as an inner product of the form $k(u, v) =< \phi(u), \phi(v) >$ in an embedding feature space determined by the map ϕ which needs not be given explicitly. Applications of kernel methods to graphs require the construction of graph kernels, i.e., functions that are capable of measuring similarity between two data instances of the graph. In graph kernels, the distance between two data instances is computed by counting common substructures (e.g., walks, paths, and trees) in the KG (Ristoski & Paulheim, 2016b).

Lösch *et al.* (2012) introduced two graph kernels especially suited for RDF, based on intersection graphs and intersection trees. The intersection graph of two graphs is a graph containing all the elements the two graphs have in common. The use of the intersection graph may become problematic as its calculation is potentially expensive: the whole instance graph for each entity has to be extracted and the two graphs have to be intersected explicitly. As an alternative, using instance trees to extract common neighborhoods of two entities enables a direct construction of the common properties, without building the instance graphs. Instances trees are obtained based on the graph expansion with respect to an entity of interest.

Later, the intersection tree path kernel was modified and simplified by De Vries & De Rooij (2013). In other works, De Vries (2013) introduced a fast approximation of the Weisfeiler-Lehman

graph kernel algorithm for RDF data. The Weisfeiler-Lehman Subtree graph kernel is a state-of-the-art kernel for graph comparison introduced by Shervashidze & Borgwardt (2010). The kernel computes the number of sub-trees shared between two (or more) graphs by using the Weisfeiler-Lehman test of graph isomorphism. This algorithm creates labels representing subtrees in a given number of iterations. They also have developed another type of kernels over RDF data, the RDF walk count kernel (De Vries & de Rooij, 2015). The random walk kernels are based on a simple idea: given a pair of graphs, perform random walks on both, and count the number of matching walks. In this work, the RDF walk count kernel counts the different walks in the sub-graphs (up to the provided graph depth) around the instances nodes. The approaches developed by Lösch *et al.* and by Vries *et al.* have been applied to two common relational learning tasks: entity classification and link prediction.

3.2 Graph Embeddings

An embedding is a vector representation resulting from the use of semantic information mapping techniques. In graph embeddings, the KG is transformed into sequences of entities, which can be considered as corpus' sentences. Then, based on the corpus, vector representations are generated using neural language models (Ristoski & Paulheim, 2016b).

Ristoski & Paulheim (2016a) proposed RDF2Vec that uses language modeling approaches for unsupervised feature extraction from sequences of words and adapts them to RDF graphs. The first step of this approach is converting the graph into a set of sequences of entities, which can be considered as sentences. Two general approaches are used for converting graphs into a set of sequences of entities: graph walks and Weisfeiler-Lehman Subtree RDF Graph Kernels. In the second step, they use those sequences of entities to train a neural language model, which estimates the likelihood of a specific sequence of entities appearing in a graph. One of the most popular and widely used is the Word2vec neural language model (Mikolov *et al.*, 2013). There are two different algorithms, the Continuous Bag-of-Words model and the Skip-Gram model. Once the training is finished, each entity in the graph is represented as a vector of latent numerical features. Projecting such latent representations of entities into a lower dimensional feature space shows that semantically similar entities appear closer to each other. This approach was applied to a number of classification and regression tasks, using two types of RDF graphs: small domain-specific RDF datasets and large cross-domain RDF datasets.

Smaili *et al.* (2018) proposed Onto2Vec that also uses language modeling approaches to generate vector representations of biological entities in ontologies by combining formal ontology axioms and annotation axioms from the ontology. An ontology is treated as a set of axioms, each of which constitutes a sentence. Word2vec methods are used to process the axioms syntactically, and the vector representations are obtained in such a way that words with similar contexts tend to be close to each other in the vector space. To evaluate Onto2Vec, GO was used to produce vector representations of the proteins, the GO classes to which they are annotated and the axioms in GO that constrain these classes. Onto2Vec was then applied to PPI prediction on different datasets and the identification of protein families.

3.3 Semantic Similarity

The use of SS in the biomedical field is recent, however, there are already a wide variety of bioinformatics applications that benefit from using SSMs over biomedical KGs, namely: PPI prediction (Jain & Bader, 2010; Lin *et al.*, 2004; Liu *et al.*, 2018; Maetschke *et al.*, 2011; Patil & Nakamura, 2005; Wu *et al.*, 2006; Zhang & Tang, 2016), prediction of disease-associated genes (Freudenberg & Propping, 2002; Li *et al.*, 2014; Liu *et al.*, 2018; Perez-Iratxeta *et al.*, 2002; Turner *et al.*, 2003; Zhang *et al.*, 2006), validation of function prediction (Duan *et al.*, 2006), network prediction (Lee & Lee, 2005), prediction of cellular localization (Lei & Dai, 2006), and automatic annotation validation (Couto *et al.*, 2006).

Given the popularity of GO, several approaches explore the SS over the GO KG to compare proteins based on what they do, rather than using sequence similarity. Jain & Bader (2010) proposed an algorithm, Topological Clustering Semantic Similarity, that uses the SS between GO classes annotated to proteins to distinguish true from false protein interactions. The central idea is to find subsets of GO terms defining similar concepts and score gene products belonging to a similar subset higher that if they belong to different sets. Furthermore, this algorithm considers the unequal depth of biological knowledge representation in different branches of the GO graph.

Liu *et al.* (2018) proposed a method that incorporates enrichment of GO classes by a gene pair in computing the SS. This GO enrichment was incorporated by querying gene pair in the computation of IC of a GO term. The enrichment of a GO term by the pair of genes depends on whether the term is annotated by one gene or by both genes in the pair. Thus, the probability of a GO term is defined as the joint probability of the term as inferred by background corpus and as annotated by two querying genes. The effect of introducing GO enrichment on several SSMs was tested for prediction of sequence homologies, gene expression correlations, PPIs, and disease-associated genes.

Although GO is the most widely-used biomedical ontology, other ontologies have also been used, including Human Phenotype Ontology (HPO) (Robinson *et al.*, 2008). In HPO, the ontology terms correspond to phenotypic abnormalities and are used for annotation of diseases. Köhler *et al.* (2009) presented a method for clinical diagnostics based on measure phenotypic similarity between queries and hereditary diseases annotated with HPO. This method uses the semantic structure of the HPO to weight clinical features on the basis of specificity and to identify features that, if present, best distinguish among the top candidate differential diagnoses. The semantic network defined by the HPO is used to refine the differential diagnosis. Furthermore, a statistical model was developed to assign p-values to the resulting similarity scores, which can be used to rank the candidate diseases.

Hoehndorf *et al.* (2011) developed a method to combine phenotype ontologies with anatomy ontologies and apply a measure of SS to generate PhenomeNET, a cross-species network of phenotypic similarity between genotypes and diseases. This method was used to identify orthologous genes, genes involved in the same pathway and gene-disease associations.

3.4 Other Semantic Representations

There are other approaches that explore innovative KG-based semantic representations. The tools FeGeLOD (Paulheim & Fümkranz, 2012) and RapidMiner (Ristoski *et al.*, 2015) generate data mining features based on the exploration of specific or generic relations in the graph. These approaches use different unsupervised feature generation strategies for creating new data mining features from LOD sources:

- The *Direct Types* generator extracts all types for an entity;
- The *Datatype Properties* generator extracts all datatype properties;
- The *Relations* generator creates a binary or a numeric attribute for each property that connects an entity to other entity;
- The *Qualified Relations* generator creates a binary or a numeric attribute for properties, taking the type of the related entity into account;
- The *Specific Relations* generator creates features for a user-specified relation.

All the generators are able to retrieve the hierarchical relations between the features. Since not all the features generated by the different strategies are equally helpful, a simple heuristic is applied to filter them out.

Bandyopadhyay & Mallick (2017) proposed a novel set of features to represent a protein pair using their annotated GO terms, including their ancestors. In this approach, a protein pair is treated as a bag of words, where the GO classes annotating (i.e., describing) the two proteins represent the words. The feature value of each word is calculated using the IC of the corresponding term multiplied by a coefficient, which represents the weight of that term inside a protein pair. To evaluate this approach, the authors tested the performance of supervised classifiers like Random Forest and SVM to predict PPIs using the proposed feature vectors.

Maetschke *et al.* (2011) used GO-driven algorithms for PPI inference introducing the concept of inducers. Inducers are motivated by the assumption that an induced term set is richer in information, and can be a more accurate predictor of protein interaction, than the original annotation. Term inducers define sets of GO terms that are induced within the DAG by the GO annotation of protein pairs and are subsequently projected onto a feature vector. In this work, three classes of inducers were used: (i) basic inducers that ignore term relationships and represent the traditional machine learning approach (the annotation of a protein pair is described by a vector that encodes either all assigned GO terms or the GO terms shared by the two proteins); (ii) ancestral inducers that are based on ancestor terms derived from a set of protein annotations and resemble node-based SSMs; (iii) shortest-path inducers that include terms along the shortest path or paths between two term sets and are similar to edge-based SSMs. The inducers were then used to predict protein interactions using different classifiers, such as Random Forest and Naïve Bayes.

Chen *et al.* (2019) proposed a hybrid feature representation of proteins that combines not only GO information but also protein sequence properties and interaction topology. The GO-based features characterize protein pairs based on the clustering of GO terms. One GO-based

feature is defined as one GO-term cluster indexed by a lowest common ancestor. The network-based features are derived from the topological properties (for instance the number of common neighbors) of a PPI network. This PPI network is constructed by linking the proteins with a SS above a certain threshold. The SS threshold is obtained by deriving a reference PPI network, from the training set of protein pairs. The hybrid features representations were integrated with multiple learning algorithms for PPI prediction. This ensemble learning approach adopts a stacked generalization scheme, where five classifiers were combined into one predictive unit.

Chapter 4

Methodology

This chapter gives an outline of the proposed methodology and the methods that should be used in each step. An overview of the methodology is shown in Figure 4.1.

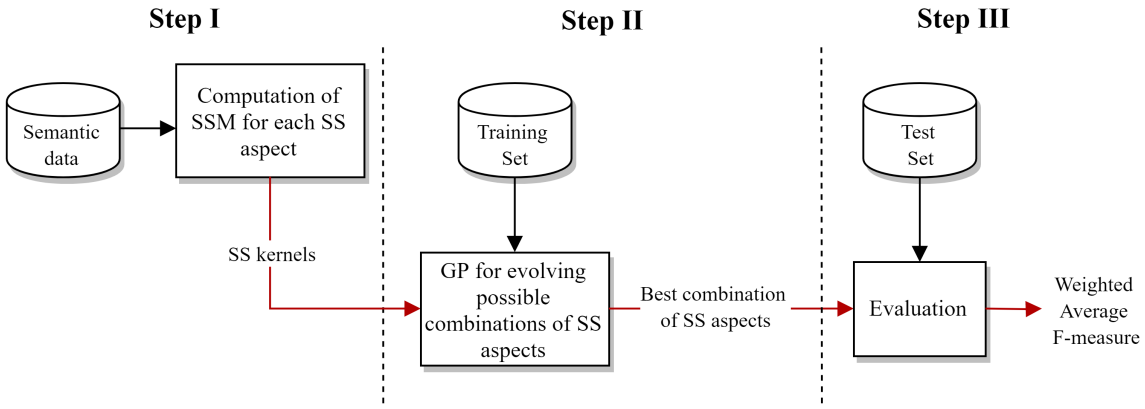

Figure 4.1: Overview of the methodology.

The guiding hypothesis of this dissertation is that GP can learn suitable combinations of SS aspects to support specific supervised classification tasks. In supervised classification, there is a set of training examples and the objective is learning how to predict the classification of unseen examples. Thus, the implementation of the proposed methodology takes as input a KG and training and test sets.

4.1 Step I: Computation of Semantic Similarity

The first step is computing the SSM for each SS aspect. The semantic representations based on SS are built by taking into account the global structure of the KG. However, KGs typically provide multiple perspectives over an entity, either by describing it using different properties or using different portions of the graph. Therefore, the SSs aspects can be either the properties or portions of the graph and the SS can be computed taking each aspect into consideration.

Relatively to the computation of SS, the rise in the number of SSMs designed for different biomedical ontologies was accompanied by the development of tools to calculate them. For example, Semantic Measures Library (SML) (Harispe *et al.*, 2013) is an open source Java library dedicated to the computation and analysis of semantic measures, such as SS. This library supports various ontology formats and specifications (e.g., OBO, RDF, OWL) and provides a large collection of semantic measures.

4.2 Step II: Evolving Combinations

The second step is using GP to learn on the training test the best combination of the different SS aspects. Here, the programs evolved by GP are the possible combinations of the SS aspects. The fitness function that guides evolution is based on the success of a given combination of SS aspects in a specific task.

Once again there are several GP applications and packages, such as gplearn[1], that implement this evolutionary algorithm. The Pyhton package gplearn extends the scikit-learn machine learning library (Pedregosa *et al.*, 2011) and is designed to solve regression problems, but can also be used for binary classification. In addition, this package allows tweaking several parameters, including the function set, fitness function, parsimony coefficient, size of initial population and number of generations.

4.3 Step III: Evaluation

The last step is the evaluation on the test set, using the evolved combination to support the supervised learning task. The performance measure is the Weighted Average F-measure (WAF). This metric accounts for class unbalance by computing the F-measure for each class and then calculating the average of all computed F-measures, weighted by the number of instances of each class:

$$\text{WAF} = \frac{\sum_{c \in C} \text{F-measure}_\text{c} \times \text{Support}_\text{c}}{\sum_{c \in C} \text{Support}_\text{c}} \tag{4.1}$$

where C is the set of classes, F-measure_c is the F-measure computed for class c, and Support_c is the number of instances in class c.

The F-measure (for a class c) is the weighted harmonic mean of the precision and recall and is given by

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.2}$$

where

[1] https://gplearn.readthedocs.io

$$\text{Precision} = \frac{\text{Number of instances correctly classified as class } c}{\text{Number of instances classified as class } c} \tag{4.3}$$

and

$$\text{Recall} = \frac{\text{Number of instances correctly classified as class } c}{\text{Number of instances labeled as class } c}. \tag{4.4}$$

A key aspect of our evaluation approach is to compare our methodology that is able to evolve a combination of semantic aspects to static combinations established *a priori*. This allows us to compare our methodology to a scenario where semantic aspects are selected and combined by experts before the learning task.

Chapter 5

Application to Protein-Protein Interaction Prediction

A major challenge in biological systems is the accurate mapping of the interactome, i.e., the set of all PPIs within a cell. PPIs are responsible for many critical functions in biology and are highly relevant to disease states. Discovering new interactions through laboratory experiments is expensive and time-consuming, so the total number of explored interactions is still very low compared to the whole proteome. For this reason, computational methods have been applied to predict PPIs.

In this chapter, we apply the methodology presented in Chapter 4 to predict PPIs. Using the GO, the most popular biomedical ontology, as KG, we analyse the methodology in nine benchmark datasets. The next sections describe the data sources and methodology implementation aspects and present the results and discussion.

5.1 Data Sources

The implementation of the proposed methodology for PPI prediction took as input an ontology file, a protein annotation file and a list of protein pairs, that are described in the following subsections.

5.1.1 Knowledge Graph

The KG used in this work is composed by the GO and GO annotations. The development of GO, the most widely-used biological ontology, was motivated by the semantic heterogeneity of biomedical data and its lack of formality. This ontology is constantly revised and expanded as biological knowledge accumulates.

GO defines the universe of classes (also called "GO terms") associated with gene product (proteins or RNA) functions and how these functions are related to each other with respect to

three aspects: (i) BP, which captures the larger process accomplished by multiple molecular activities in which the gene product is active; (ii) MF, biochemical (or molecular-level) activity of a gene product; (iii) CC, the location relative to cellular structures in which a gene product performs a function.

GO terms and their semantic relations form a hierarchical DAG where each node represents a GO term and the edges represent the relationships between the terms. The edges can represent different types of relations (e.g., *is_ a*, *part_ of*, *regulates*, *has_ part*). GO is loosely hierarchical, with "child" terms being more specialized than their "parent" terms, but unlike a strict hierarchy, a term may have more than one parent term. The ancestor terms in the hierarchy subsume the semantics of descendent terms. The three GO aspects are represented as root nodes of the graph since they are unrelated and do not have a common parent node (see Figure 2.2). These aspects are *is_ a* disjoint, so no *is_ a* relations operate between terms from the different aspects. However, other relationships such as *part_ of* and *regulates* do operate between the GO aspects. For example, the relation *part_ of* operates between the molecular function term "cyclin-dependent protein kinase activity" and the biological process term "cell cycle".

A GO annotation associates a specific gene product with a specific class in the GO, identifying some aspect of its function. For instance, in Figure 2.3 the gene product for *ACES HUMAN* is annotated with the GO class *amyloid percursor protein metabolic process*. A single gene product may be annotated with several classes across all semantic aspects of GO.

The GO graph was collected from www.geneontology.org (dated January 2019) in OBO format and contains 45006 ontology terms subdivided into 4206 CC terms, 29689 BP terms, and 11111 MF terms. Only *is-a* relations were considered. GO annotations were downloaded from Gene Ontology Annotation (GOA) database (Huntley *et al.*, 2014) (dated January 2019) for four species (*S. cerevisiae*, *H. sapiens*, *E. coli* and *D. melanogaster*) in Gene Association File (GAF) 2.1 format. These annotations link Uniprot (Apweiler *et al.*, 2004) identifiers for proteins with GO classes describing them.

5.1.2 Benchmark Protein-Protein Interaction Datasets

For evaluation and comparison, we used benchmark PPI datasets of different species. These datasets were produced by other works and have been applied by several others in evaluating PPI approaches. Table 5.1 provides the number of interactions and the author for each dataset.

The positive data (interacting protein pairs) of these datasets were collected from existing databases. The negative data is obtained by random sampling of protein pairs since experimental high-quality negative data (non-interacting protein pairs) is hardly available. Random sampling is based on the assumption that the expected number of negatives is several orders of magnitude higher than the number of positives, such that the negative space is randomly sampled with larger probability than the positive space (Park, 2009). In most of the datasets, negative data is generated by randomly creating protein pairs that are not reported to interact. In the dataset GRID/HPRD-bal-HS a different strategy is employed to achieve balanced random sampling. Here, the number of times each protein appears in the negative set is equal to the number of times it appears in the positive set, with the negative set still being composed of protein pairs that are not known to interact.

Table 5.1: PPI Benchmark Datasets with their authors and numbers of total interactions (I), positive interactions (PI) and negative interactions (NI). The STRING-SC, STRING-HS, STRING-EC, STRING-DM datasets are available in http://bioinformatics.org.au/tools/go2ppi/#training. The DIP-HS dataset is available in http://baderlab.org/Software/TCSS. The BIND-SC and DIP/MIPS-SC datasets are available in https://noble.gs.washington.edu/proj/sppi/. The GRID/HPRD-bal-HS and GRID/HPRD-unbal-HS datasets are available in http://www.bioinformatics.leeds.ac.uk/BRS-nonint/PPI_RandomBalance.html.

Dataset	Author	I	PI	NI
STRING-SC	Maetschke *et al.* (2011)	30476	15238	15238
STRING-HS	Maetschke *et al.* (2011)	6980	3490	3490
STRING-EC	Maetschke *et al.* (2011)	2334	1167	1167
STRING-DM	Maetschke *et al.* (2011)	642	321	321
DIP-HS	Jain & Bader (2010)	2826	1391	1435
BIND-SC	Ben-Hur & Noble (2005)	1499	749	750
DIP/MIPS-SC	Ben-Hur & Noble (2005)	14498	4825	9673
GRID/HPRD-bal-HS	Yu *et al.* (2010)	31608	15804	15804
GRID/HPRD-unbal-HS	Yu *et al.* (2010)	31608	15804	15804

Given the evolving nature of GO annotations, some benchmark proteins are no longer found in current GOA files. Consequently, we removed all pairs that failed to meet this criterion: both proteins have at least one annotation in one semantic aspect. Furthermore, the yeast datasets do not use UniProt identifiers. We used the Protein Identifier Cross-reference (PICR) tool (Côté *et al.*, 2007) web application to map protein identifiers to the corresponding UniProt accession numbers. PICR provides programmatic access through Representational State Transfer (REST) that is very useful since we simply need to build a well-formatted RESTful URL. Thus, not all identifiers could be mapped to UniProt and those proteins were removed. Table 5.2 provides the species and the number of interactions for each dataset after excluding the pairs that did not meet the above criteria.

Table 5.2: PPI benchmark datasets, with number of positive interactions (PI) and number of negative interactions (NI) after exclusion.

Dataset	Species	PI	NI
STRING-SC	*S. cerevisiae*	15218	15166
STRING-HS	*H. sapiens*	3460	3452
STRING-EC	*E. coli*	1127	1118
STRING-DM	*D. melanogaster*	288	262
DIP-HS	*H. sapiens*	1375	1364
BIND-SC	*S. cerevisiae*	749	750
DIP/MIPS-SC	*S. cerevisiae*	4659	9148
GRID/HPRD-bal-HS	*H. sapiens*	15675	15674
GRID/HPRD-unbal-HS	*H. sapiens*	15675	15645

5.2 Methodology Implementation

Figure 5.1 shows the implementation of the proposed methodology for PPI prediction. The following subsections refer to the implementation of each step of the methodology.

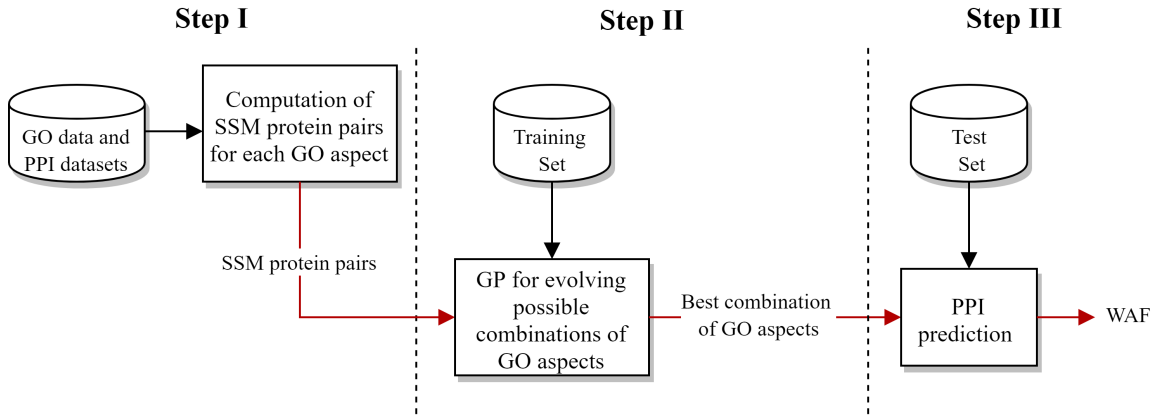

Figure 5.1: Implementation of the proposed methodology for PPI prediction.

5.2.1 Semantic Similarity Measures

For PPI prediction using GO, the semantic aspects are the three GO aspects. Thus, the SSs corresponding to each semantic aspect were computed for each protein pair in our input data.

Given the popularity of GO, SSMs have been extensively proposed and studied using this ontology as a source of knowledge. In this dissertation, six different SSMs were employed, summarized in Table 5.3.

Table 5.3: Summary of SSMs used to calculate the SS between proteins.

SSM	IC	Type of approach	Techniques
SimGIC/IC_{Seco}	Intrinsic	graph-based	Jaccard
$Resnik_{Max}$/IC_{Seco}	Intrinsic	best pairs	Maximum
$Resnik_{BMA}$/IC_{Seco}	Intrinsic	best pairs	Average
SimGIC/IC_{Resnik}	Extrinsic	graph-based	Jaccard
$Resnik_{Max}$/IC_{Resnik}	Extrinsic	best pairs	Maximum
$Resnik_{BMA}$/IC_{Resnik}	Extrinsic	best pairs	Average

Each SSM includes two approaches: the approach used to calculate the IC of each GO term (IC_{Seco} or IC_{Resnik}); the IC-based approach used to calculate the similarity between two sets of GO terms (SimGIC or $Resnik_{Max}$ or $Resnik_{BMA}$).

IC_{Seco} is a structure-based approach proposed by Seco *et al.* (2004) based on the number of

direct and indirect descendants and given by

$$\mathrm{IC}_{\mathrm{Seco}}(t) = 1 - \frac{\log\left[\mathrm{hypo}(t) + 1\right]}{\log\left[\mathrm{maxnodes}\right]} \tag{5.1}$$

where $\mathrm{hypo}(t)$ is the number of direct and indirect descendants from term t (including term t) and maxnodes is the total number of concepts in the ontology.

$\mathrm{IC}_{\mathrm{Resnik}}$ is a corpus-based approach proposed by Resnik (1995) and based on a corpus of GO annotations of all gene-products in an organism, which is given by

$$\mathrm{IC}_{\mathrm{Resnik}}(t) = -\log p(t) \tag{5.2}$$

where $p(t)$ is the probability of annotation in the corpus.

SimGIC is a groupwise approach proposed by Pesquita *et al.* (2007), based on a Jaccard index in which each GO term is weighted by its IC and given by

$$\mathrm{simGIC}(p_1, p_2) = \frac{\sum_{t \in \{\mathrm{GO}(p_1) \cap \mathrm{GO}(p_2)\}} \mathrm{IC}(t)}{\sum_{t \in \{\mathrm{GO}(p_1) \cup \mathrm{GO}(p_2)\}} \mathrm{IC}(t)} \tag{5.3}$$

where $\mathrm{GO}(p_i)$ is the set of annotations (direct and inherited) for protein p_i.

$\mathrm{Resnik}_{\mathrm{Max}}$ and $\mathrm{Resnik}_{\mathrm{BMA}}$ are pairwise approaches based on the class-based measure proposed by Resnik (1995) in which the similarity between two classes corresponds to the IC of their most informative common ancestor. This pairwise approach is used with two combination variants, maximum

$$\mathrm{Resnik}_{\mathrm{Max}}(p_1, p_2) = \max\left\{\mathrm{sim}(t_1, t_2) : t_1 \in \mathrm{GO}(p_1), t_2 \in \mathrm{GO}(p_2)\right\} \tag{5.4}$$

and best-match average

$$\mathrm{Resnik}_{\mathrm{BMA}}(p_1, p_2) = \frac{\sum_{t_1 \in \mathrm{GO}(p_1)} \mathrm{sim}(t_1, t_2)}{2|\mathrm{GO}(p_1)|} + \frac{\sum_{t_2 \in \mathrm{GO}(p_2)} \mathrm{sim}(t_1, t_2)}{2|\mathrm{GO}(p_2)|} \tag{5.5}$$

where $|\mathrm{GO}(p_i)|$ is the number of annotations for protein p_i and $\mathrm{sim}(t_1, t_2)$ is the SS between the GO term t_1 and GO term t_2 and is defined as

$$\mathrm{sim}(t_1, t_2) = \max\left\{\mathrm{IC}(t) : t \in \{\mathrm{A}(t_1) \cap \mathrm{A}(t_2)\}\right\} \tag{5.6}$$

where $\mathrm{A}(t_i)$ is the set of ancestors of t_i.

These measures were selected because SimGIC and $\mathrm{Resnik}_{\mathrm{BMA}}$ represent high-performing group and pairwise approaches in predicting sequence, Pfam and Enzyme Commission similarity (Pesquita *et al.*, 2007), whereas $\mathrm{Resnik}_{\mathrm{Max}}$ helps to elucidate whether a single source of similarity is enough to establish an interaction. With regard to IC measures, $\mathrm{IC}_{\mathrm{Resnik}}$ measure relies on the frequency of each annotation in the corpus of GO annotations for all gene-products in an organism depending on the size and nature of input corpus, while $\mathrm{IC}_{\mathrm{Seco}}$ measure only relies on

the GO hierarchical structure. These measures are representative of the two main approaches for IC calculation and help to understand if the scope and size of the GO annotations are adequate and large enough to provide accurate IC calculations.

The SML was employed to support SS calculations. The implementation of SML dedicated to GO requires GO in OBO format and the protein annotations in GAF 2.0. Since the most recent GO annotation file is in GAF 2.1 format, it was converted to the older format specifications.

After SS computations, each instance of the datasets, that represents a pair of proteins, was characterized by three values, corresponding to the SS between them for the three GO aspects, and a label (interact or non-interact).

5.2.2 Genetic Programming and Supervised Learning

For PPI prediction, the models evolved by GP are simply combinations of the SS of the three aspects. GP evolves a good (hopefully the best) combination of the different SS aspects to support PPI prediction. Figure 5.2 shows a parse tree of one of the simplest combinations evolved in our experiments, here translated as

$$\max(BP, CC) \times \max(BP, MF) \tag{5.7}$$

where the SS aspects BP, CC and MF are the variables.

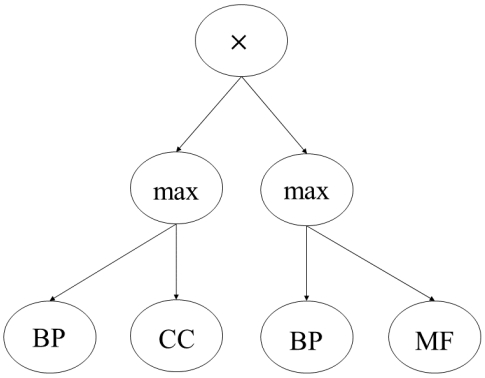

Figure 5.2: Combination generated by GP for PPI prediction. Max stands for Maximum.

The gplearn package was employed to implement GP. A "vanilla" tree-based GP system was used, with no extras to boost the performance. The parameters are listed in Table 5.4. All others were used with the default values of the gplearn software. The parsimony coefficient is a non-standard parameter, specific to gplearn, and consists of a constant that penalizes large programs by adjusting their fitness to be less favorable for selection. It was set to 10^{-5}, a value experimentally found to reduce the size of the evolved models without compromising their fitness. The function set contained only the four basic arithmetic operators ($+$,$-$,$\times$, and $\div$, protected against division by zero as in Koza (1992)), plus the Maximum (max) and Minimum (min) operators. Although there is a vast array of tunable parameters even in the most basic

Table 5.4: GP parameters for PPI prediction.

Parameter	Value
Number of generations	50
Size of population	500
Function set	$+,-,\times,\div$,max,min
Fitness function	RMSE
Parsimony coefficient	10^{-5}

GP system, normally they do not substantially influence the outcome in terms of best fitness achieved (Sipper *et al.*, 2018).

For binary classification, it is fairly standard to use GP in a regression-like fashion, where the expected class labels are treated as numeric expected outputs, and the fitness function that guides the evolution is based on the error between the expected and predicted values. Therefore, we used this same system in our experiments, with the root mean square error (RMSE) as fitness function. However, when we report the performance of the evolved models, we first transform the real-valued predicted outputs in class labels, by applying a cutoff value. If the predicted value is higher than 0.5, the predicted label is 1 (interaction), otherwise the predicted label is 0 (no interaction).

For the purpose of cross validation, GP learnt the models on a training set, and their performance was then assessed on a test set. Before each run of GP, the original dataset was split into training and test sets using a 70–30 ratio by stratified subset selection. In this context, stratification means that the random split returns training and test subsets that have the same proportions of class labels as the original dataset.

5.2.3 Performance Measure

Since GP is a stochastic process, in each experiment we performed 10 runs, splitting the dataset into a new 70-30 partition in every run. At the end of each run, we evaluated the WAF of classifications on the respective test set using the combination selected by GP. We report the performance of GP as the median of the 10 obtained WAFs.

As baselines, we have used five static combinations:

- The three single aspects (BP, MF or CC);
- Two well-known strategies for combining the single aspect scores: the Average and Maximum of the single aspect scores.

To establish the performance of these baselines, the prediction of PPI was formulated as a classification problem where a SS score for a protein pair exceeding a certain threshold (SS cutoff) indicates a positive interaction. The SS threshold was chosen after evaluating the WAF at different threshold intervals and selecting the maximum. This emulates the best choice that

a human expert could theoretically select. By comparing the performance of these optimal baselines to the performance of our proposed approach, we aim at investigating the ability of GP to learn combinations of semantic aspects that are able to support improved classification performance.

5.3 Results and Discussion

In this section, the results of this work are presented, discussed and compared with other published related work. First, the results for PPI prediction for each benchmark dataset using static combinations (corresponding to baselines) and using the proposed methodology are described. Then, the results obtained using the proposed methodology with different combinations of datasets for training and testing are presented. The different combinations of datasets allow doing intra-species, cross-species and multi-species test sets, addressing the limitations of predicting PPI for small datasets and species with fewer known interactions. Lastly, the main results of the GP models analysis and those from relevant related work are compiled.

5.3.1 Static Combinations

Prior to performing the comparative evaluation, we investigated the behavior of the different SS approaches employed, coupled with the five baselines.

Figures 5.3 to 5.11 show the WAF of classification at different cutoffs with six SSMs for the DIP-HS, STRING-HS, GRID/HPRD-unbal-HS, GRID/HPRD-bal-HS, BIND-SC, DIP/MIPS-SC, STRING-SC, STRING-DM, STRING-EC PPI datasets, by this order.

While Figure 5.3 is representative of the behavior found for the other datasets, Figure 5.11 shows a different behavior, where the F-measure is less penalized at higher cutoffs, particularly for the Maximum and CC results. The proteins in this dataset have fewer BP annotations, which may help explain the improved performance of CC.

Comparing the charts for different SSMs, we observe that the results obtained using IC_{Seco} (SimGIC/Ic_{Seco}, $Resnik_{Max}$/Ic_{Seco}, $Resnik_{BMA}$/Ic_{Seco}) do not seem to be significantly different from their homologues (SimGIC/Ic_{Resnik}, $Resnik_{Max}$/Ic_{Resnik}, $Resnik_{BMA}$/Ic_{Resnik}). However, for each set of curves for SimGIC, $Resnik_{Max}$ and $Resnik_{BMA}$ approaches, the maximum F-measure is achieved at different ranges of SS cutoff. For SimGIC (Figure 5.3-a and 5.3-d), $Resnik_{Max}$ (Figure 5.3-b and 5.3-e) and $Resnik_{BMA}$ (Figure 5.3-c and 5.3-f) the ranges are approximately $[0.1-0.3]$, $[0.6-0.8]$ and $[0.2-0.5]$, respectively. For most datasets, each SSM shows a consistent behavior with curves having similar shapes.

Furthermore, we verify that the maximum observed F-measure is achieved when Resnik approaches are used. The differences between SSMs are not unexpected since SimGIC considers multiple GO annotations for calculating SS while Resnik approaches only consider the best-matching term pairs. Therefore, the better performance using Resnik approaches makes sense because proteins in PPIs only need to be in proximity in a single location or participate in a single shared biological process, to be biologically relevant for PPI prediction.

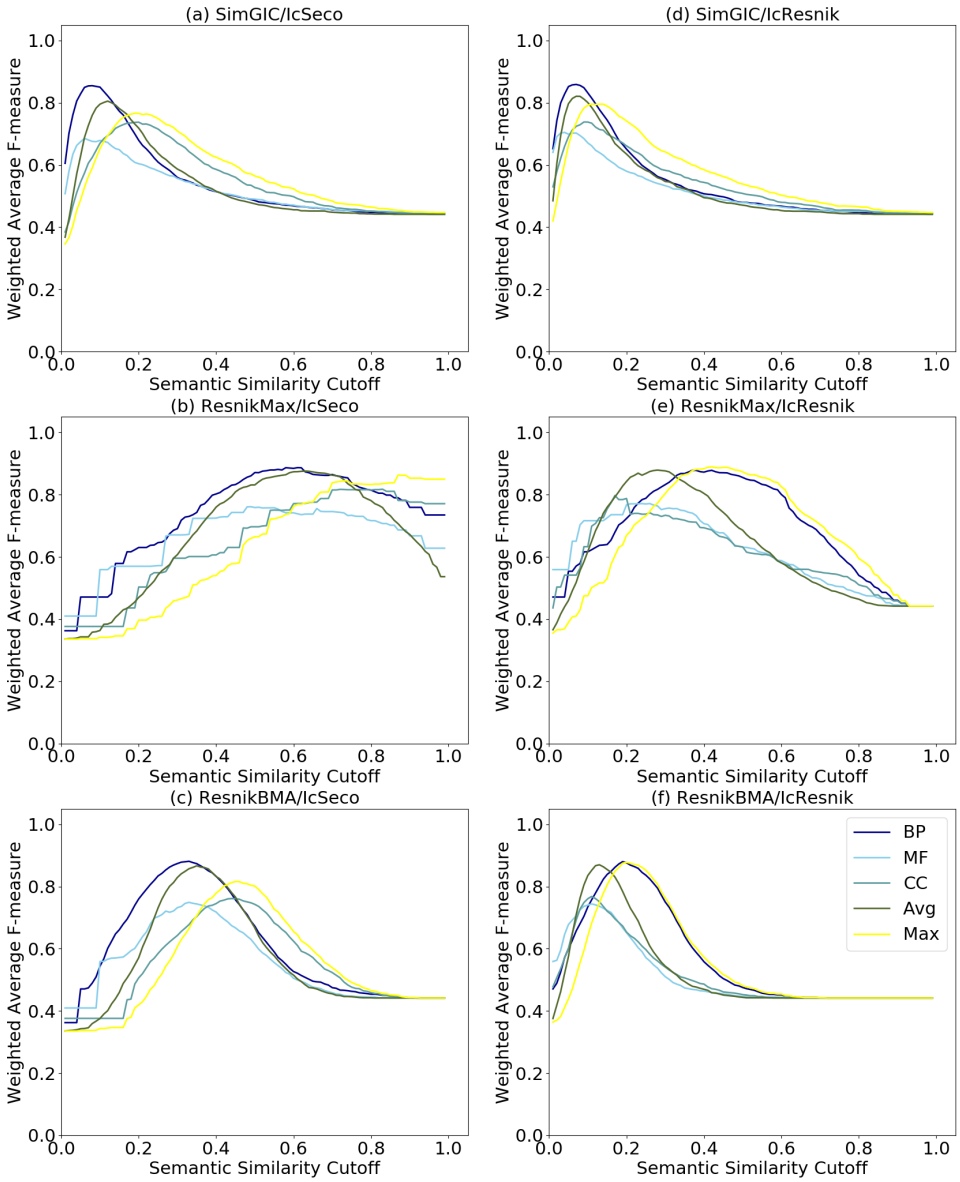

Figure 5.3: WAF curves for DIP-HS PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

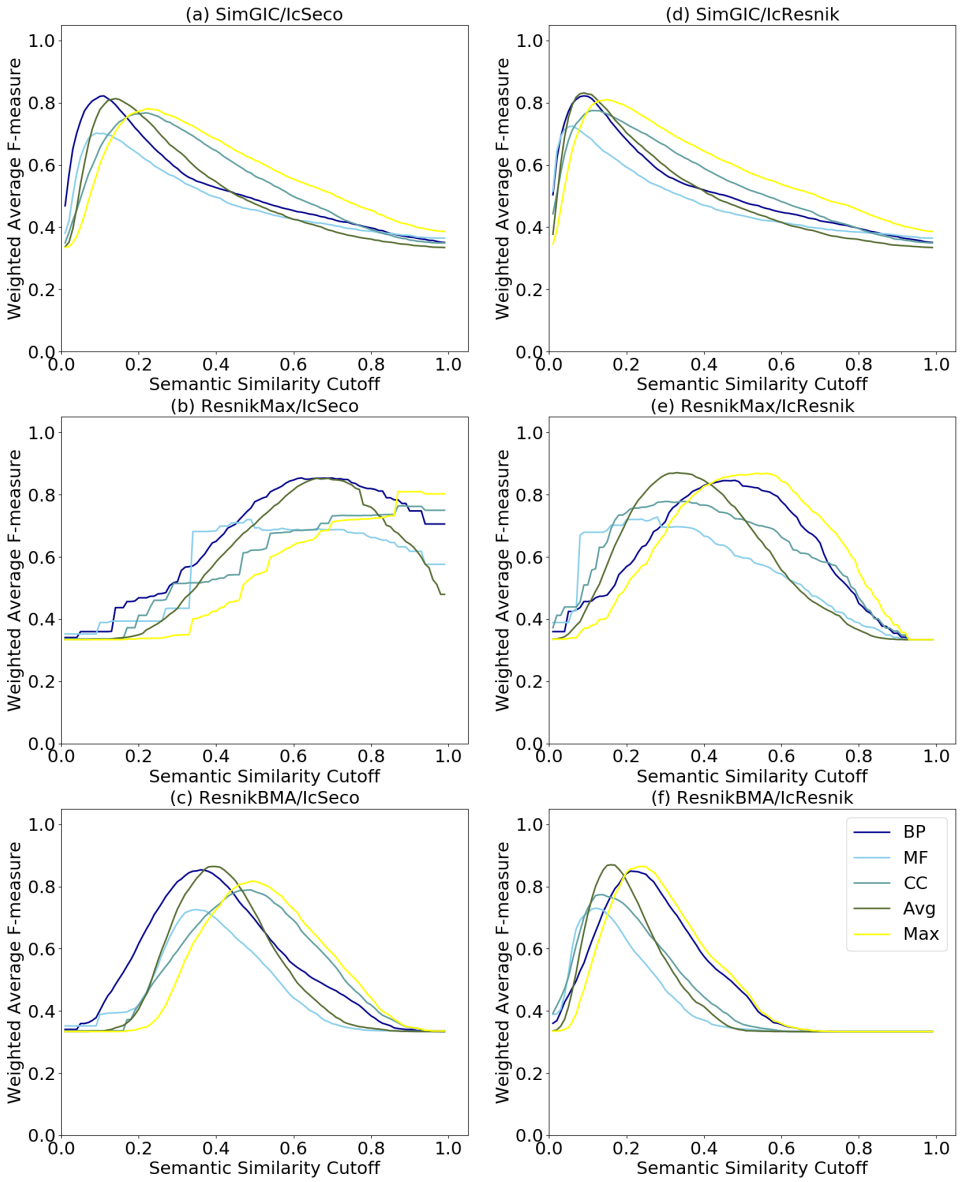

Figure 5.4: WAF curves for STRING-HS PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

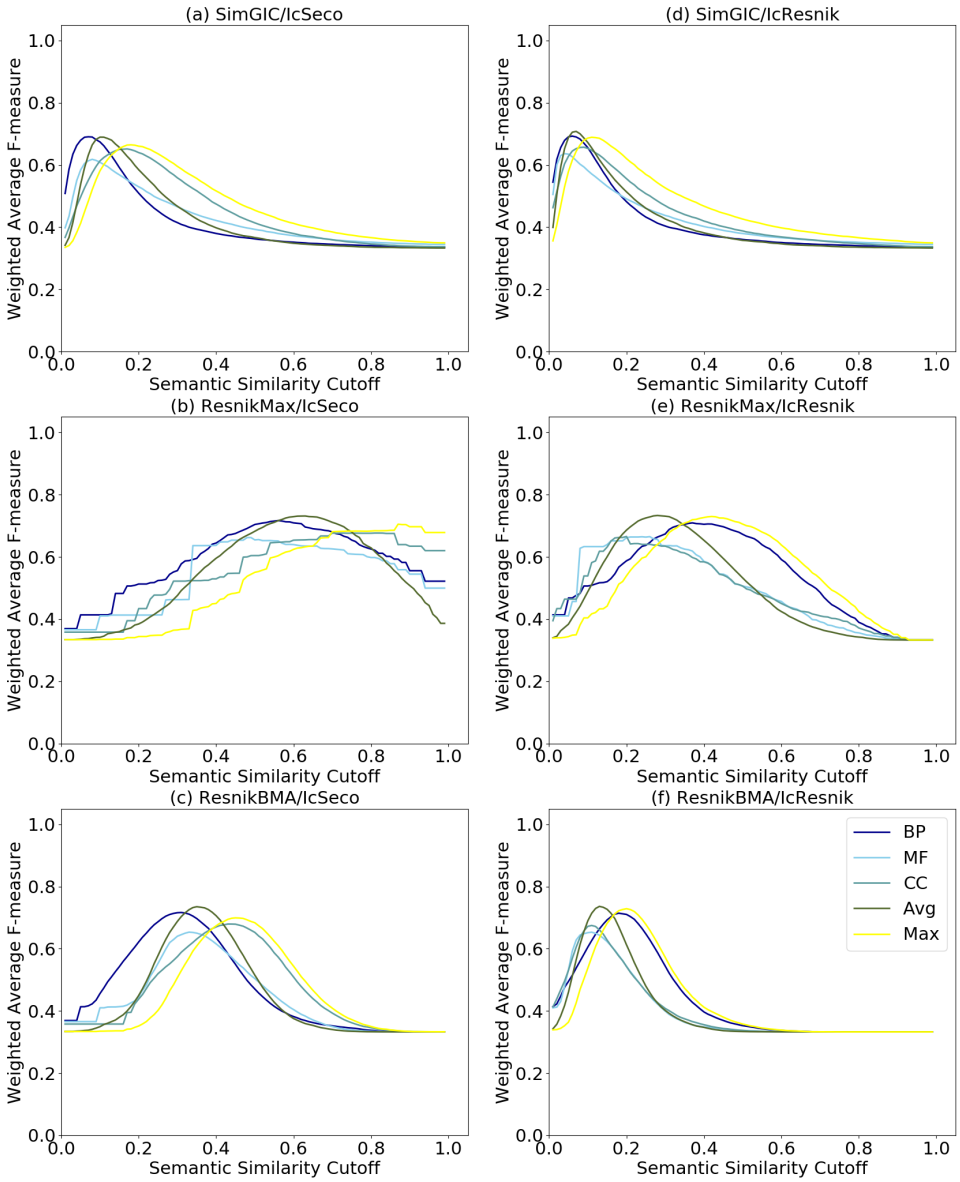

Figure 5.5: WAF curves for GRID/HPRD-unbal-HS PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

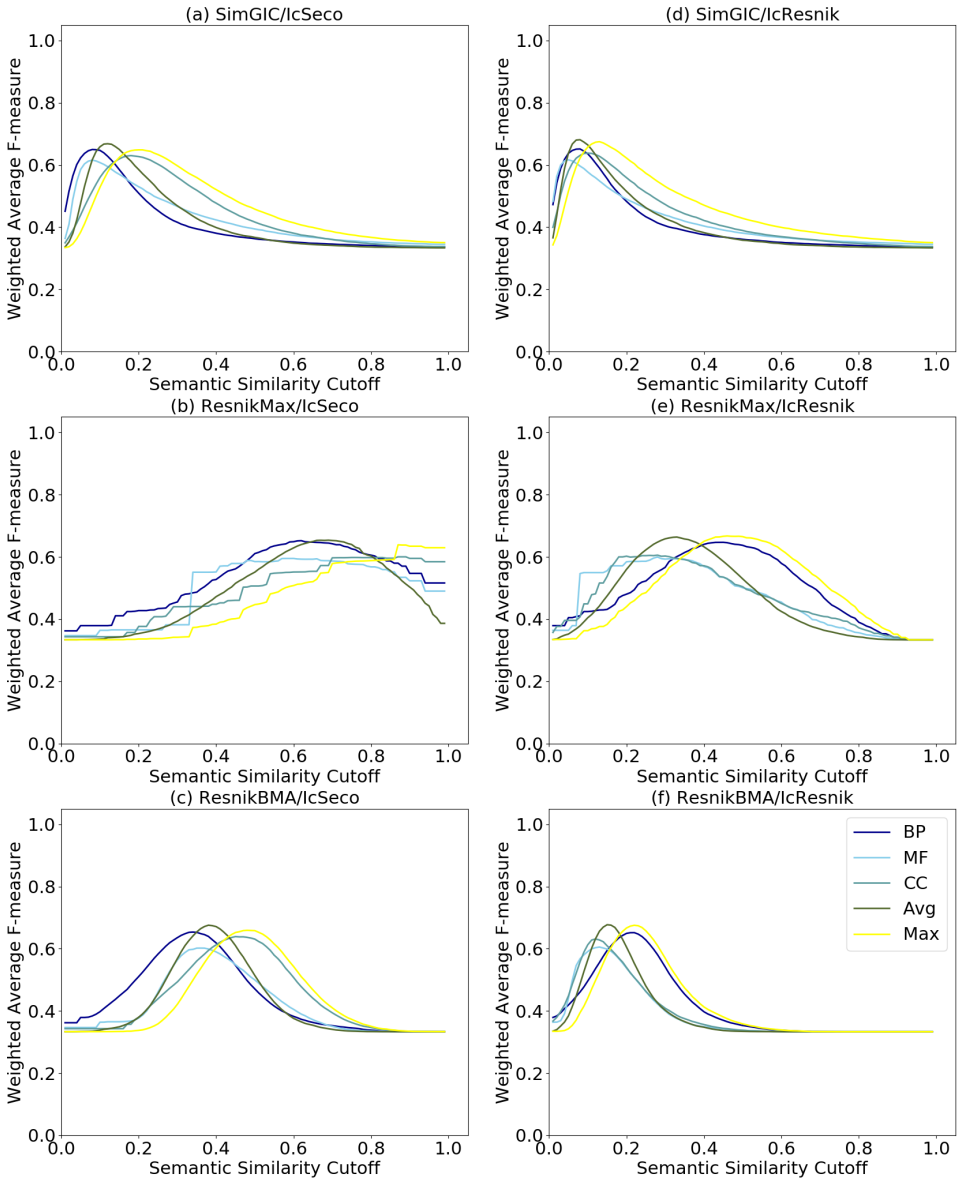

Figure 5.6: WAF curves for GRID/HPRD-bal-HS PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

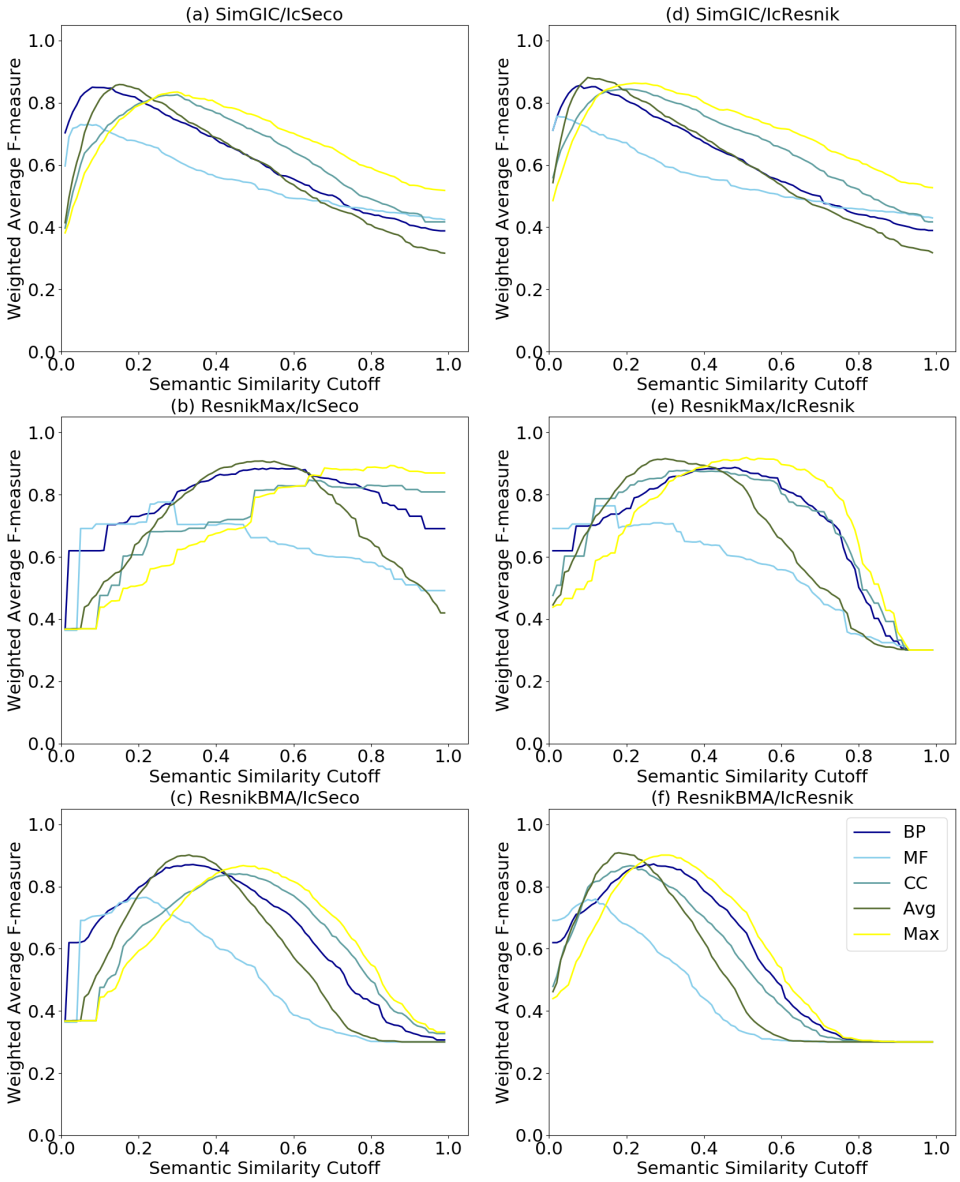

Figure 5.7: WAF curves for BIND-SC PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

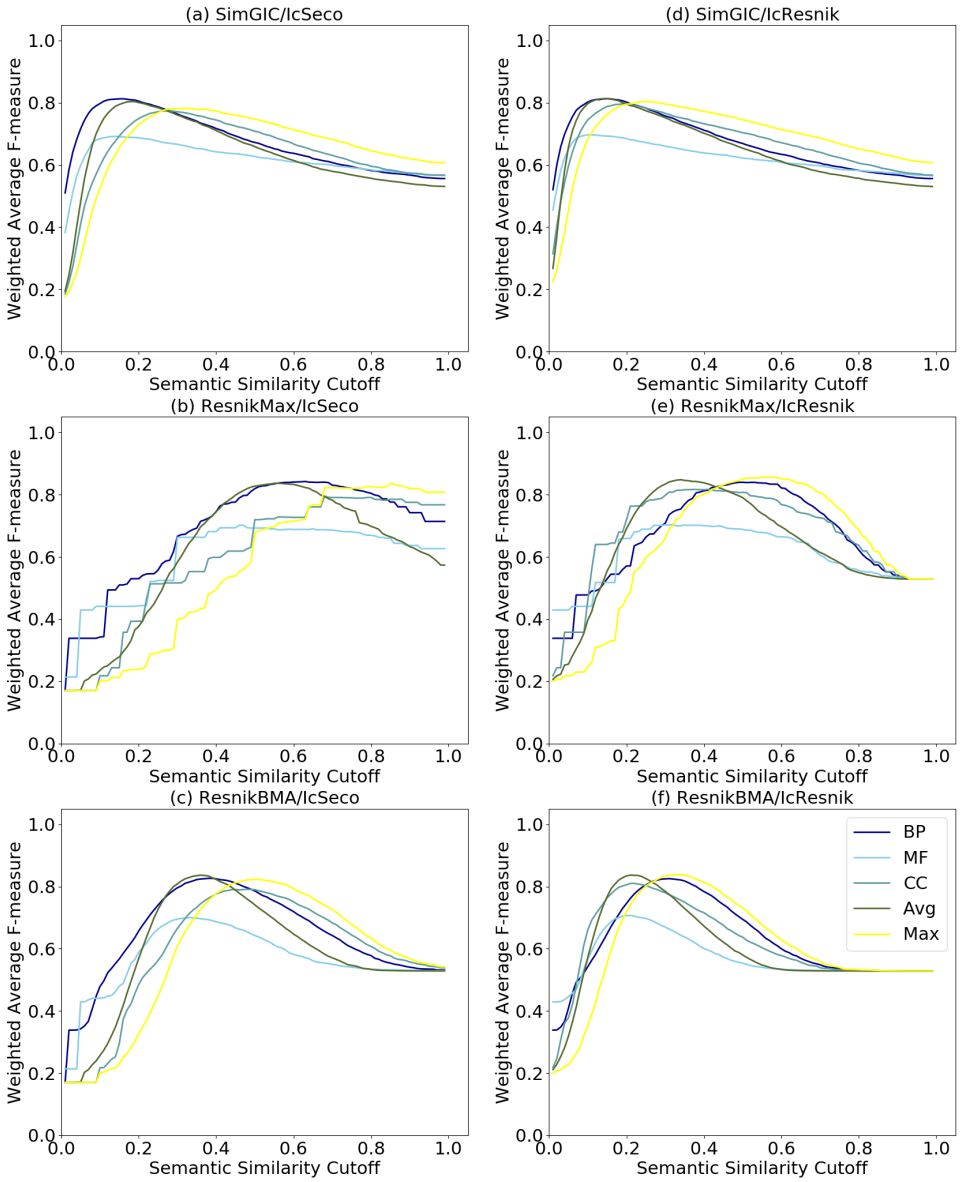

Figure 5.8: WAF curves for DIP/MIPS-SC PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

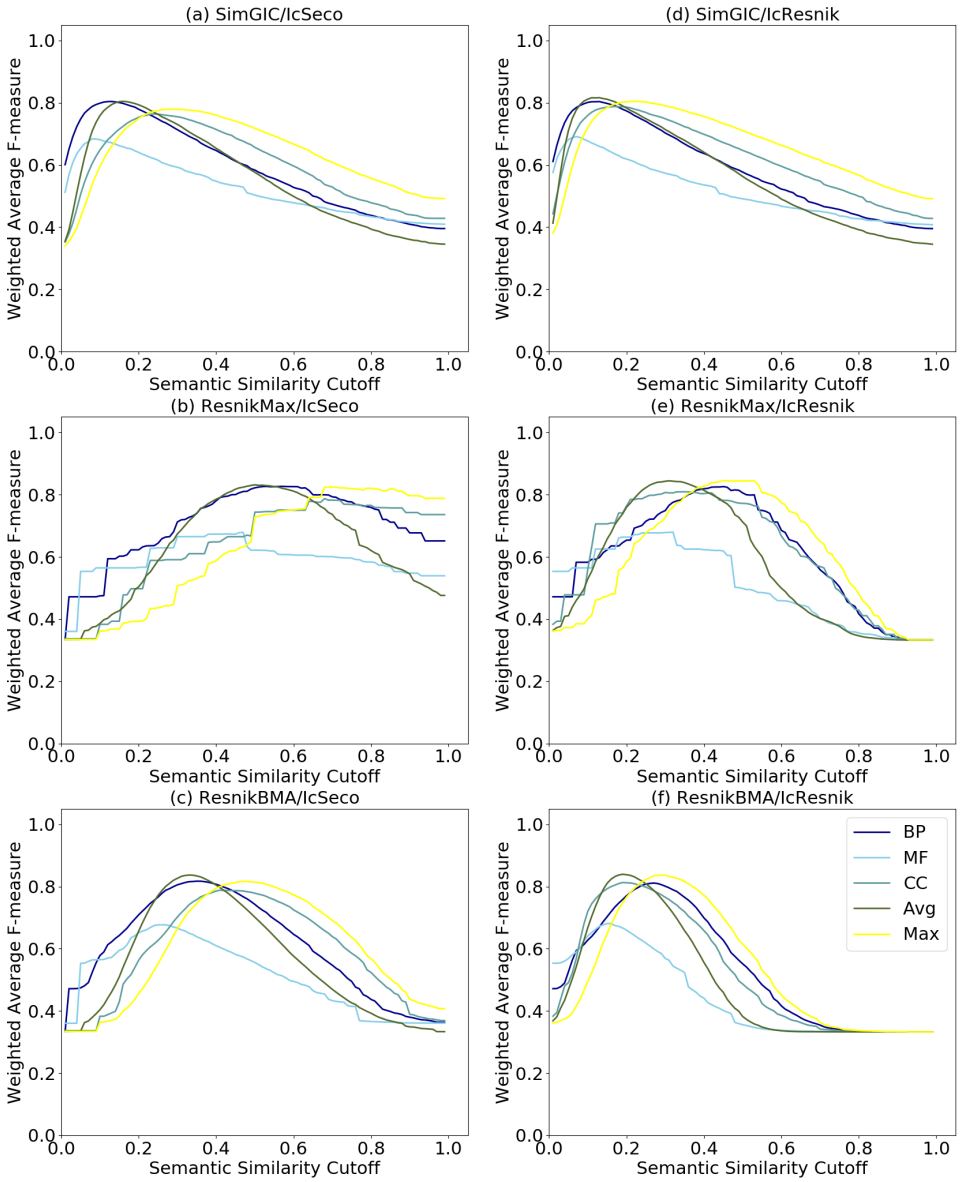

Figure 5.9: WAF curves for STRING-SC PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

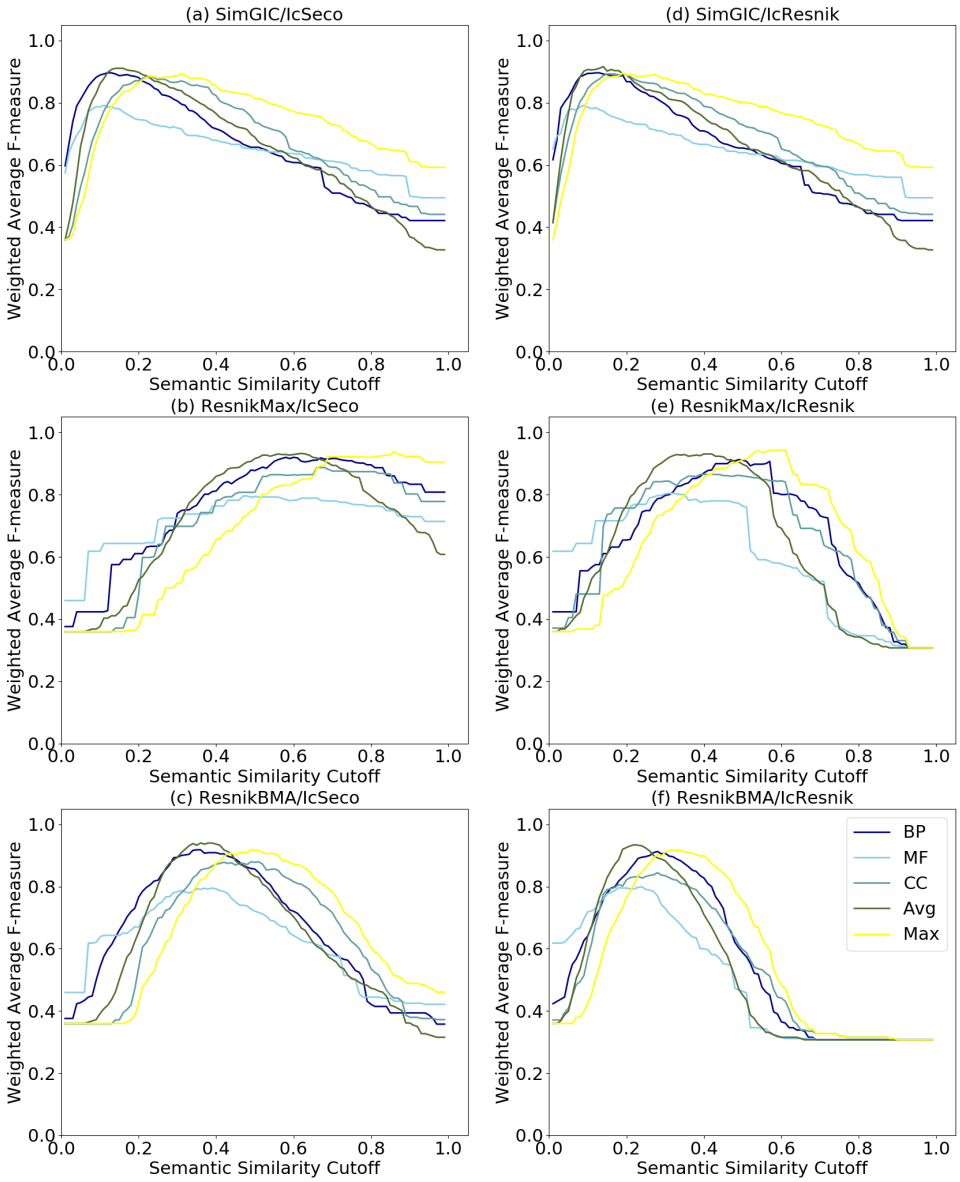

Figure 5.10: WAF curves for STRING-DM PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

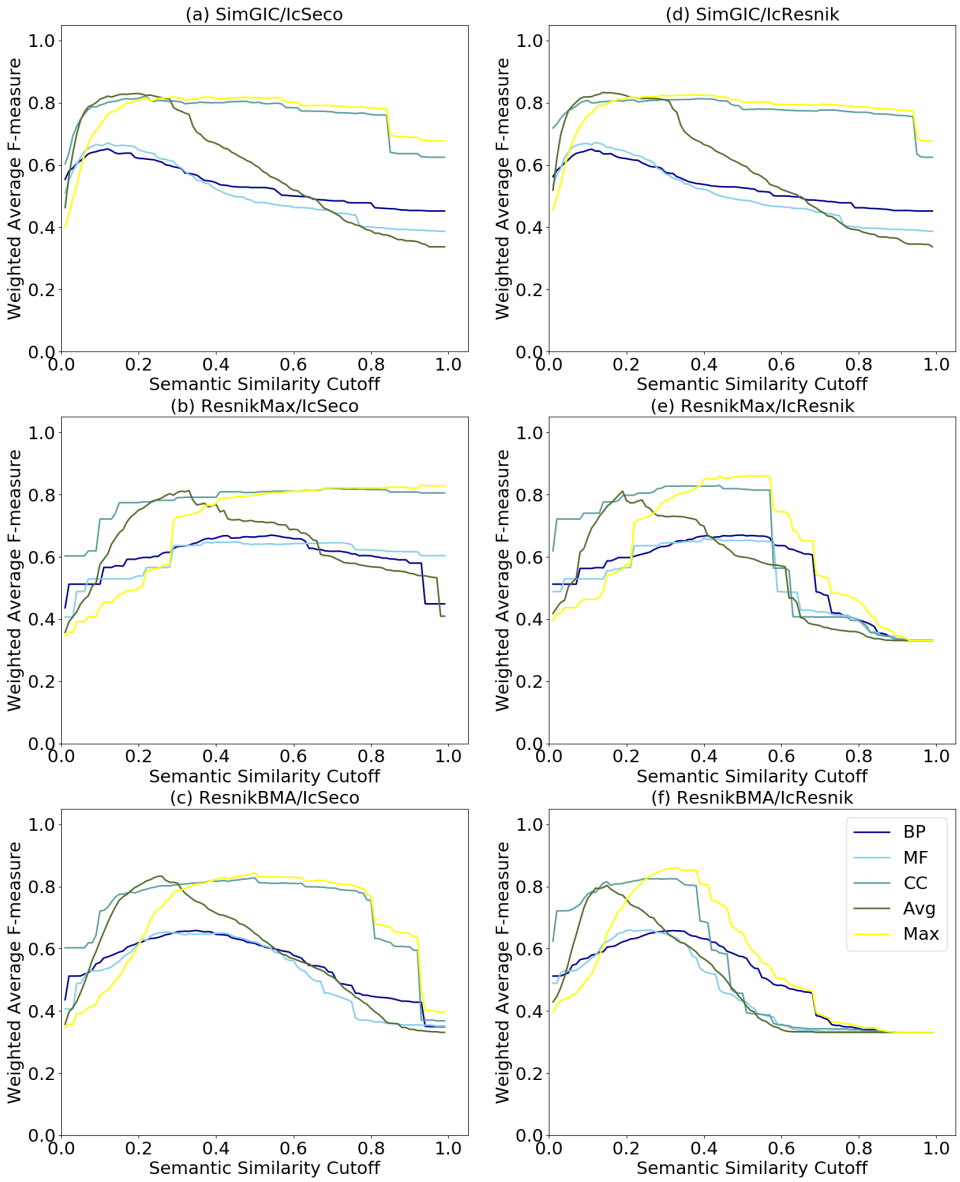

Figure 5.11: WAF curves for STRING-EC PPI dataset. WAF evaluations with static combinations of semantic aspects (CC, BP, MF, Avg and Max) at different cutoffs are shown. The evaluation is performed using six SSMs: (a) SimGIC/Ic_{Seco}, (b) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Seco}}$, (c) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Seco}}$, (d) SimGIC/$\text{Ic}_{\text{Resnik}}$, (e) $\text{Resnik}_{\text{Max}}/\text{Ic}_{\text{Resnik}}$ and (f) $\text{Resnik}_{\text{BMA}}/\text{Ic}_{\text{Resnik}}$.

The maximum F-measure achieved in each baseline is presented in Table 5.5. As expected, these results indicate that the predictive power of the BP and CC aspects is similar, with a slight advantage for BP, while the predictive power of MF is considerably lower. The dataset STRING-EC (Figure 5.11) is an exception because using only the SS for BP ontology provides worse results comparatively to the other combinations of single aspects. Once again, the explanation for that can be the lack of BP annotations for the species *E. coli*. Regarding static combination approaches, the Average combination outperforms the Maximum in most cases. This is possibly due to the fact that the Average combination can take into consideration both the BP and the CC aspects.

5.3.2 Evolved Combinations

Table 5.5 shows the maximum WAF of classification for the baselines and the median of WAFs for the proposed methodology, using different SSMs.

In seven out of nine datasets, GP is able to learn combinations of semantic aspects that improve the best classification performance obtained by the baselines for that dataset. In DIP/MIPS-SC and BIND-SC datasets, GP is unable to improve the performance obtained by the Maximum static combination. While our approach for DIP/MIPS-SC dataset achieves 0.2% lower performance, for BIND-SC the differences between WAF values are more significant (1.1% lower performance). However, BIND-SC is one of the smallest (less than 2000 protein pairs), which may help explain the lower performance of our approach.

Improvements over the single aspect baselines are, as expected, more pronounced for MF (up to 18%) than for the other aspects. The improvements are also clear when considering the combination baselines (2-5% in most cases). However, GP is not able to improve performance for all SSMs and sometimes shows worse performance than the Average and Maximum static combinations. It is important to note that the baselines are built to emulate the scenario of a researcher choosing an optimal threshold and employing two well-known strategies for combining the single aspect scores. With GP, we have always used the 0.5 cutoff with no further tuning, and have used a function set that included the Maximum but not the Average (which interestingly did not guarantee success or failure when compared to these two baselines).

It is interesting to note as well, that often GP achieves its best WAF when used with $\text{Resnik}_{\text{Max}}$ approaches (in six out of nine datasets). $\text{Resnik}_{\text{Max}}$ approaches, more specifically the $\text{Resnik}_{\text{Max}}/\text{IC}_{\text{Seco}}$ measure, is also the best overall measure for the single aspect baselines. For that reason, in the following experiments, the results were obtained using only $\text{Resnik}_{\text{Max}}/\text{IC}_{\text{Seco}}$ as SSM.

Finally, the comparison of results using datasets obtained with different methods of selecting negative examples can be relevant. It is known that the preparation of negative data for PPI prediction is an important issue for training and assessing the performance of a classifier. Since there are no "gold standard" non-interactions, different methods are used for choosing negative examples for training a predictor of PPIs. Some authors suggest that the negative set should include protein pairs constructed from different cellular locations or involved in different biological processes, based on the observation that pairs of proteins that have different localization patterns are unlikely to interact (Ben-Hur & Noble, 2006). However, this type of negative data

Table 5.5: Maximum WAF of classifications with baseline methodologies and the median of WAFs with the evolved combinations (EC) for the different PPI datasets. In bold, the best result for each dataset-SSM pair. In underlined, the best result for each dataset.

Dataset (#interactions)	SSM	Single and Static combinations					EC
		BP	CC	MF	Avg	Max	
STRING-EC (2245)	$SimGIC/IC_{Seco}$	0.652	0.822	0.671	0.830	0.819	**0.834**
	$Resnik_{Max}/IC_{Seco}$	0.670	0.820	0.649	0.813	0.832	**0.859**
	$Resnik_{BMA}/IC_{Seco}$	0.659	0.828	0.654	0.833	**0.844**	0.840
	$SimGIC/IC_{Resnik}$	0.651	0.813	0.672	**0.833**	0.825	0.832
	$Resnik_{Max}/IC_{Resnik}$	0.670	0.830	0.660	0.812	0.861	**0.862**
	$Resnik_{BMA}/IC_{Resnik}$	0.658	0.826	0.660	0.804	**0.860**	0.845
STRING-DM (550)	$SimGIC/IC_{Seco}$	0.896	0.886	0.792	0.911	0.893	**0.912**
	$Resnik_{Max}/IC_{Seco}$	0.920	0.887	0.796	0.933	**0.938**	0.930
	$Resnik_{BMA}/IC_{Seco}$	0.918	0.880	0.795	0.940	0.918	**0.952**
	$SimGIC/IC_{Resnik}$	0.896	0.893	0.792	**0.916**	0.896	0.915
	$Resnik_{Max}/IC_{Resnik}$	0.913	0.865	0.804	0.931	**0.944**	0.933
	$Resnik_{BMA}/IC_{Resnik}$	0.913	0.845	0.799	**0.935**	0.918	0.933
BIND-SC (1366)	$SimGIC/IC_{Seco}$	0.850	0.826	0.731	0.859	0.835	**0.868**
	$Resnik_{Max}/IC_{Seco}$	0.885	0.847	0.776	**0.909**	0.894	0.906
	$Resnik_{BMA}/IC_{Seco}$	0.871	0.841	0.765	**0.902**	0.868	0.898
	$SimGIC/IC_{Resnik}$	0.856	0.844	0.757	0.881	0.863	**0.888**
	$Resnik_{Max}/IC_{Resnik}$	0.888	0.878	0.764	0.916	**0.920**	0.910
	$Resnik_{BMA}/IC_{Resnik}$	0.873	0.867	0.760	**0.909**	0.901	0.903
DIP/MIPS-SC (13807)	$SimGIC/IC_{Seco}$	0.813	0.774	0.692	0.804	0.782	**0.817**
	$Resnik_{Max}/IC_{Seco}$	0.843	0.795	0.702	0.837	0.836	**0.848**
	$Resnik_{BMA}/IC_{Seco}$	0.826	0.791	0.700	**0.837**	0.824	0.836
	$SimGIC/IC_{Resnik}$	0.813	0.797	0.697	0.814	0.804	**0.826**
	$Resnik_{Max}/IC_{Resnik}$	0.840	0.817	0.704	0.848	**0.856**	0.854
	$Resnik_{BMA}/IC_{Resnik}$	0.825	0.810	0.707	0.837	**0.839**	0.838
STRING-SC (30384)	$SimGIC/IC_{Seco}$	0.804	0.764	0.685	0.805	0.780	**0.815**
	$Resnik_{Max}/IC_{Seco}$	0.827	0.787	0.679	0.832	0.824	**0.845**
	$Resnik_{BMA}/IC_{Seco}$	0.817	0.788	0.677	0.837	0.817	**0.840**
	$SimGIC/IC_{Resnik}$	0.804	0.788	0.691	0.816	0.805	**0.826**
	$Resnik_{Max}/IC_{Resnik}$	0.826	0.809	0.680	0.845	0.845	**0.852**
	$Resnik_{BMA}/IC_{Resnik}$	0.811	0.813	0.681	0.839	0.836	**0.844**
DIP-HS (2739)	$SimGIC/IC_{Seco}$	0.855	0.738	0.686	0.805	0.767	**0.868**
	$Resnik_{Max}/IC_{Seco}$	0.887	0.818	0.761	0.876	0.863	**0.910**
	$Resnik_{BMA}/IC_{Seco}$	0.881	0.761	0.749	0.867	0.816	**0.898**
	$SimGIC/IC_{Resnik}$	0.859	0.738	0.705	0.821	0.796	**0.862**
	$Resnik_{Max}/IC_{Resnik}$	0.878	0.797	0.771	0.879	0.889	**0.894**
	$Resnik_{BMA}/IC_{Resnik}$	0.880	0.768	0.744	0.870	0.878	**0.891**
STRING-HS (6912)	$SimGIC/IC_{Seco}$	0.822	0.768	0.703	0.813	0.781	**0.836**
	$Resnik_{Max}/IC_{Seco}$	0.854	0.764	0.720	0.853	0.811	**0.877**
	$Resnik_{BMA}/IC_{Seco}$	0.854	0.789	0.726	0.865	0.817	**0.877**
	$SimGIC/IC_{Resnik}$	0.822	0.775	0.726	0.832	0.810	**0.847**
	$Resnik_{Max}/IC_{Resnik}$	0.846	0.778	0.728	0.871	0.868	**0.882**
	$Resnik_{BMA}/IC_{Resnik}$	0.850	0.774	0.730	0.870	0.865	**0.874**
GRID/HPRD-unbal-HS (31320)	$SimGIC/IC_{Seco}$	0.691	0.652	0.618	0.690	0.665	**0.707**
	$Resnik_{Max}/IC_{Seco}$	0.717	0.677	0.662	0.731	0.705	**0.736**
	$Resnik_{BMA}/IC_{Seco}$	0.717	0.680	0.653	0.735	0.699	**0.742**
	$SimGIC/IC_{Resnik}$	0.693	0.657	0.637	0.708	0.689	**0.711**
	$Resnik_{Max}/IC_{Resnik}$	0.709	0.666	0.665	0.733	0.730	**0.736**
	$Resnik_{BMA}/IC_{Resnik}$	0.714	0.675	0.653	0.736	0.729	**0.740**
GRID/HPRD-bal-HS (31349)	$SimGIC/IC_{Seco}$	0.650	0.630	0.616	0.668	0.649	**0.669**
	$Resnik_{Max}/IC_{Seco}$	0.652	0.600	0.595	0.654	0.639	**0.661**
	$Resnik_{BMA}/IC_{Seco}$	0.654	0.639	0.602	0.676	0.659	**0.686**
	$SimGIC/IC_{Resnik}$	0.652	0.638	0.617	0.681	0.675	**0.682**
	$Resnik_{Max}/IC_{Resnik}$	0.647	0.606	0.600	0.664	0.667	**0.670**
	$Resnik_{BMA}/IC_{Resnik}$	0.652	0.631	0.606	0.677	0.676	**0.687**

selection criteria can make the dataset biased, especially when GO based features are used for PPI prediction. Other authors use a simpler schema, selecting non-interacting pairs uniformly at random. This selection scheme also has potential pitfalls because the interaction network is not complete and the negative set can be contaminated with interacting proteins. Furthermore, according to Yu *et al.* (2010), if in datasets some proteins appear many more times in the positive set than in the negative set, then a machine learning method will learn this and predict positive interactions preferentially for these proteins, which inflates classification accuracy. To address this bias, Yu *et al.* propose a method to create a balanced negative set. In this dissertation, we use two datasets, GRID/HPRD-unbal-HS and GRID/HPRD-bal-HS, with equal positive sets but with two types of negative sets, namely, random and balanced random (see Section 5.1.2). Analyzing our results, we conclude that using a negative set constructed by balanced sampling reduced the performance. However, the explanation of Yu *et al.* does not apply to our approach because our machine learning method does not "see" the proteins, it only has access to SSs.

5.3.3 Evolved Combinations for Intra-species Prediction

The previous results suggest that having fewer instances can hinder the ability of GP to learn a suitable combination of aspects. Therefore, and since two of the species have several datasets, we tested our methodology using combined sets for each of these species. This allows us to investigate whether a species-oriented model based on more instances can improve on the performance of individual datasets. The human combined set contains the data from 4 datasets (STRING-HS, DIP-HS, GRID/HPRD-bal-HS, GRID/HPRD-unbal-HS), with a total of 54219 protein pairs. The yeast combined set contains the data from three datasets (STRING-SC, BIND-SC, and DIP/MIPS-SC), with a total of 42330 protein pairs. Some pairs of proteins appear in more than one dataset so, in these combined sets, the repeated pairs are first removed from the combined sets and only then randomly split into training and test sets. Figure 5.12 shows the WAF boxplot for the three yeast datasets, the four human datasets, the yeast combined set and the human combined set. Each box includes the WAFs obtained in 10 runs.

Given the influence of the large proportion of instances coming from the larger datasets, we were expecting that the performance using the combined set was similar to the performance of the largest datasets included in the combined set. Using the boxplots to compare the prediction performance, we verified that, for yeast data, the performance of the yeast combined set is in fact very similar to the performance of the STRING-SC dataset (the largest dataset). However, for human data, the performance of the combined set is higher than the performance of the two largest human datasets, GRID/HPRD-unbal-HS and GRID/HPRD-bal-HS, suggesting that the data from smaller datasets enhances the performance of predictions.

We were also interested in investigating, within a species, the performance of training in a given group of datasets and testing on a different one. Once again, to solve the problem of repeated pairs, we determine that if a protein pair is simultaneously in the training set and in the test set, it will be removed from one of them. Tables 5.6 and 5.7 present the different tests we conducted, indicating for each test which datasets are in the training set and which are in the test set for human and yeast data, respectively.

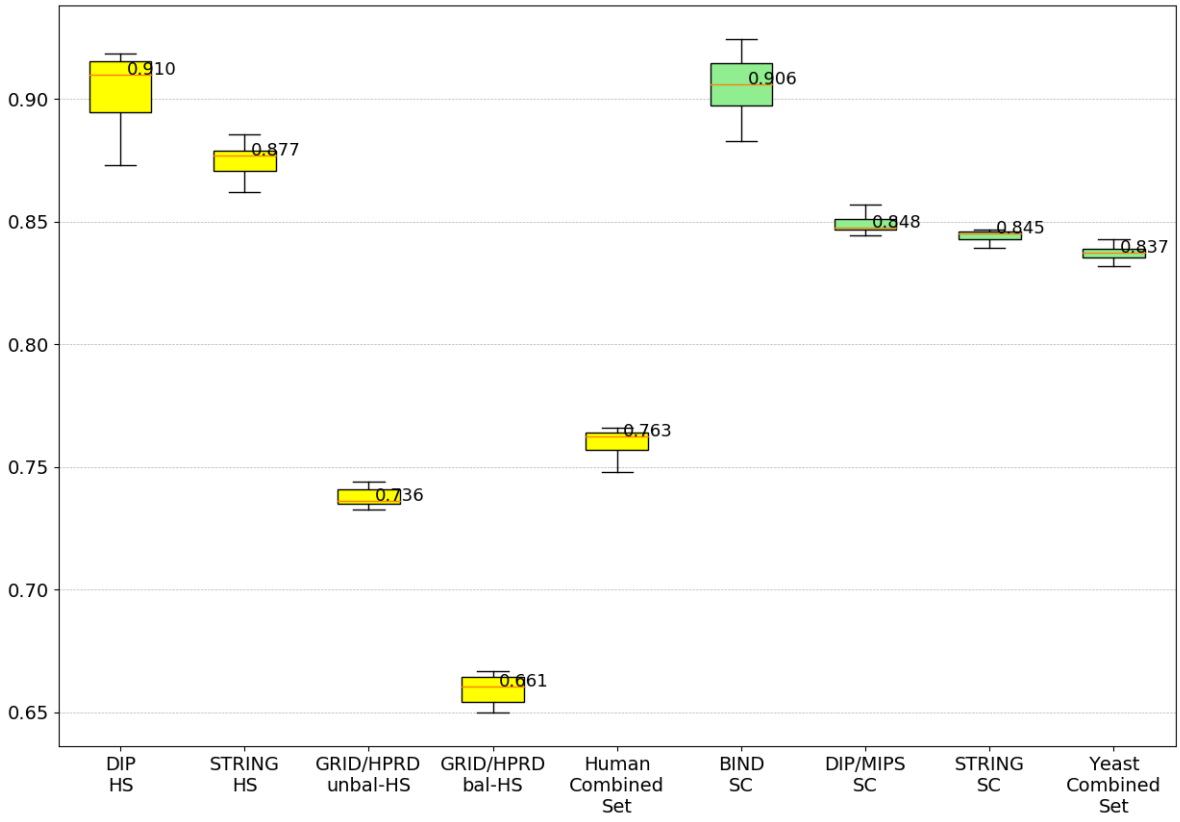

Figure 5.12: WAF boxplot using combined sets. The yellow boxes represent the WAF of predictions for human data and the green boxes represent the WAF of predictions for yeast data. Within the same species, the datasets appear on the x-axis in ascending order of size. The median of the WAF values is on top of each box.

The results for human and yeast are summarized in Figures 5.13 and 5.14, respectively. Analyzing the results for human sets, we conclude that using a larger dataset for training can improve the performance of classification. For instance, training with data from GRID/HPRD-bal-HS (e.g., S+Gb_D+Gub), the larger dataset, leads to higher test WAFs, while training with fewer data points (e.g., D_S+Gub+Gb) leads to lower WAF values. Relatively to yeast sets, the same behavior is observed. For instance, in S+D_B, the experiment with the largest training set and smallest test set, WAF is more than 5% higher than in the second best performing case.

Comparing the results in Figures 5.13 and 5.14, which illustrate the performance obtained when training and testing with different combination of datasets within the same species, with the results in Table 5.5, we verify that prediction methods are always more effective when trained and tested with the same dataset than when trained with other datasets of the same species. This is not surprising, considering how easy it is for biases to be unintentionally included in a dataset, and how much of these biases can be captured and used by a powerful method like GP,

Table 5.6: Training and test sets and number of protein pairs respectively used in each experiment. The names of the datasets STRING-HS, DIP-HS, GRID/HPRD-unbal-HS, and GRID/HPRD-bal-HS are abbreviated to S, D, Gub and Gb, respectively.

Training Set	No. of pairs	Test Set	No. of pairs
S	6912	D+Gub+Gb	47307
D	2739	S+Gub+Gb	51480
Gb	31349	D+S+Gub	22870
Gub	31320	D+S+Gb	22899
S+Gb+Gub	69581	D	2115
D+Gb+Gub	65408	S	5037
S+D	9651	Gb+Gub	44929
Gb+Gub	62669	S+D	7239
S+Gb	38261	D+Gub	17746
D+Gub	34059	S+Gb	20697
S+Gub	38232	D+Gb	17771
D+Gb	34088	S+Gub	20668

Table 5.7: Training and test sets and number of protein pairs respectively used in each experiment. The names of the datasets STRING-SC, BIND-SC, and DIP/MIPS-SC are abbreviated to S, B, and D, respectively.

Training Set	No. of pairs	Test Set	No. of pairs
S	30384	B+D	11946
D	13807	S+B	28523
B	1366	S+D	40964
S+B	31750	D	11163
S+D	44191	B	713
B+D	15173	S	27639

as long as they help achieve a good performance. Potential sources of bias could be a direct result of the scientific process, where determining the interaction of proteins is likely to target proteins that are more abundant (Bloom & Adami, 2003) or that participate in relevant processes, e.g., resistance/susceptibility to disease or stress conditions.

5.3.4 Evolved Combinations for Cross-species Prediction

In the above analysis, the training and test data come from the same species. However, training prediction methods on one species' data and testing them on another species' protein pairs may be useful to explore, since GO annotation is designed to be species independent (Ashburner *et al.*, 2000). To test this idea, we used our methodology to predict PPI but, using one species' data to train the model and another species' data to test it. Figure 5.15 displays the self-test and cross-species-test WAF boxplot using four datasets (STRING-DM, STRING-EC, STRING-HS,

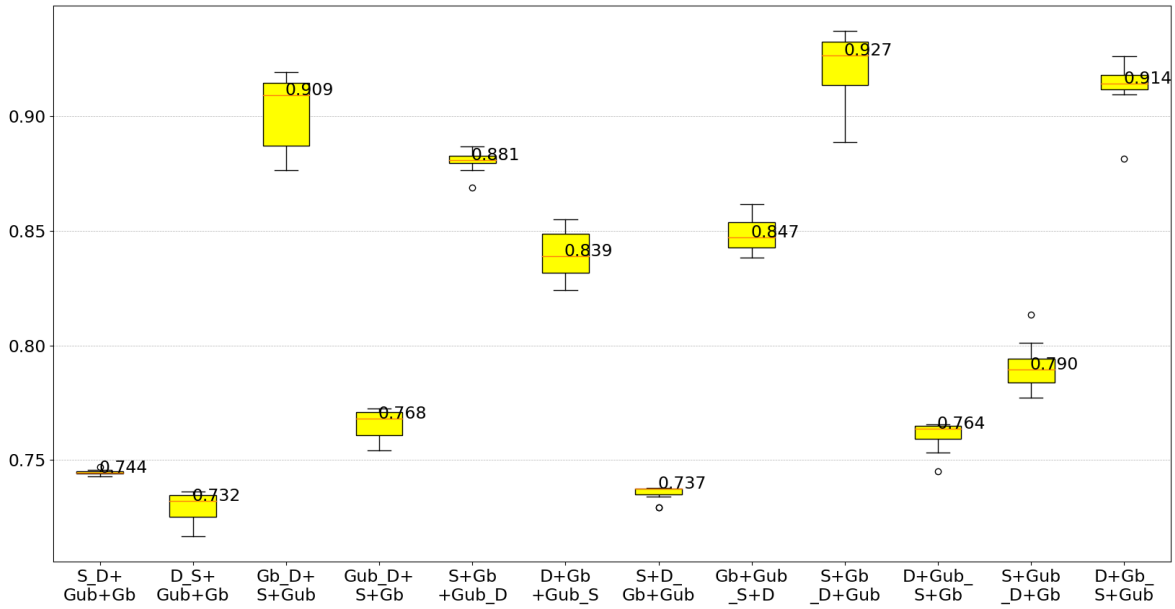

Figure 5.13: WAF boxplot using human datasets to training and testing. The labels of the plots are in format 'D1+D2_D3+D4', where D1, D2, D3, D4 are the original datasets, D1+D2 is the training set that contains data from D1 and D2, and D3+D4 is the test set that contains data from D3 and D4. In the labels, the names of the datasets STRING-HS, DIP-HS, GRID/HPRD-unbal-HS, and GRID/HPRD-bal-HS are abbreviated to S, D, Gub, and Gb, respectively.

STRING-SC) of four different species (see datasets in Table 5.2).

The results reveal that our methodology is generally more effective when trained and tested using data from the same species than when trained with data from one species and tested with data from another species. For *D. melanogaster*, performances are very similar across training sets, with *S. cerevisiae* data slightly improving performance. For *E. coli*, performance can differ greatly, with the human training set decreasing performance by more than 20% when compared to *E. coli*. In fact, training with human data gives consistently the worst results. This could be a result of the human dataset being composed of proteins that bear a lower similarity to those in other species datasets or of differences in the annotation process.

Park (2009) and Maetschke *et al.* (2011) also evaluated the cross-species accuracy by training a sequence-based classifier on one species data and predicting interactions for another species. Park found that datasets typically used for training prediction methods contain peculiar biases that limit the general applicability of prediction methods trained with them. In strong contrast, Maetshke *et al.* conclude that datasets linked to low self-test accuracy result in low cross-species accuracies while datasets with high self-test accuracy indicate datasets of good quality and, consequently, lead to high test accuracies for all training sets. This means that, according to Maetshke *et al.*, the prediction performance on the test species for different training species largely depends on the self-test accuracy achieved on the test dataset and only to a lesser degree on the training dataset. Interestingly, the results for our methodology do not seem to indicate

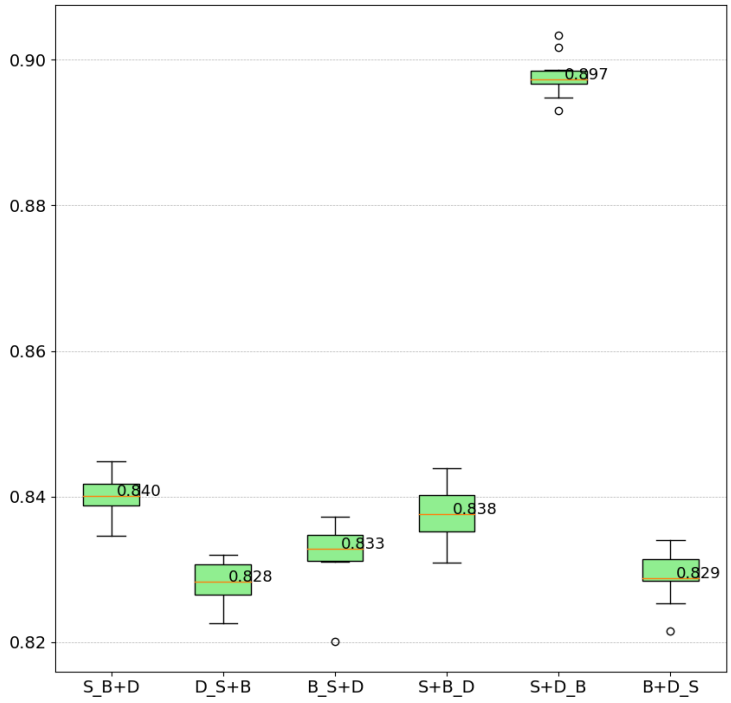

Figure 5.14: WAF boxplot using yeast datasets to training and testing. The labels of the plots are in format D1+D2_D3+D4, where D1, D2, D3, D4 are the original datasets, D1+D2 is the training set that contains data from D1 and D2, and D3+D4 is the test set that contains data from D3 and D4. In the labels, the names of the datasets STRING-SC, BIND-SC, and DIP/MIPS-SC are abbreviated to S, B, and D, respectively.

that datasets with high self-test WAF (such as STRING-DM) lead to high test WAF for all training sets.

5.3.5 Evolved Combinations for Multi-species Prediction

Applying a model learnt in more than one species data to the classification of another species data could also potentially yield interesting results. The use of diverse training data will likely produce more generally applicable models.

Therefore, we tested our approach by training the model using all species data except the one species that was used for testing. Additionally, we also ran a species-agnostic experiment where the data from all datasets was combined into a single dataset which was then randomly split into training and test sets. The strategy to remove repeated pairs used before in evolved combinations species-oriented is applied. In Figure 5.16 we can observe some interesting effects. For *D. melanogaster* and *S. cerevisiae*, the differences observed between training with the other species or with the same species are rather small: *D. melanogaster* multiple species performance

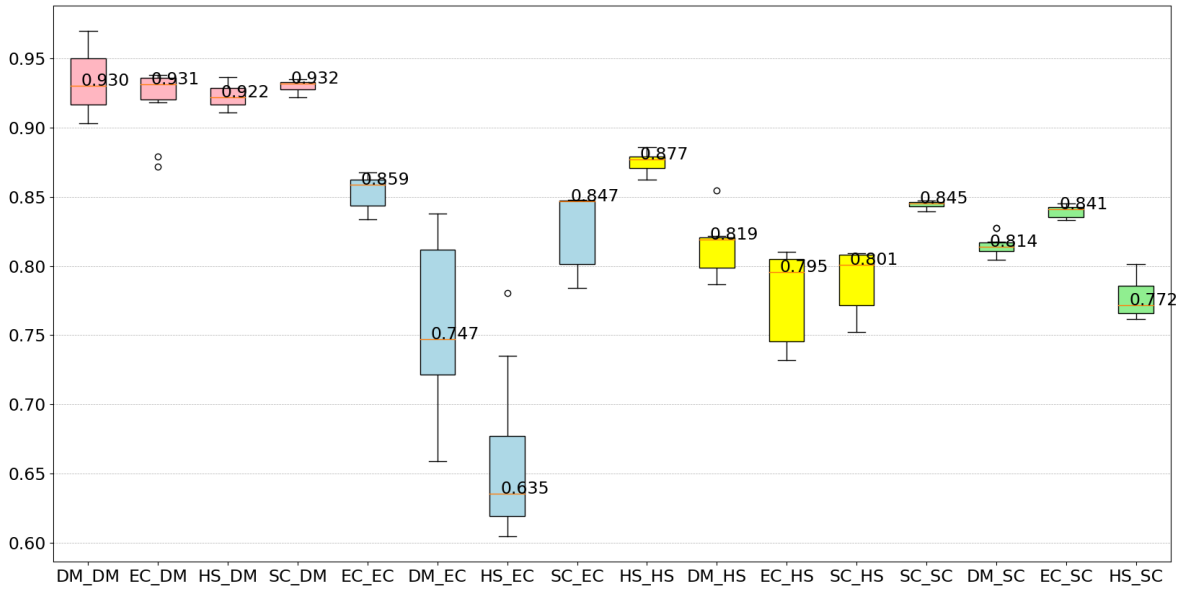

Figure 5.15: WAF boxplot using one species to train and another species to test. The format D1_D2 of the labels means training with D1 and testing on D2.

increases by 0.5%, whereas for *S. cerevisiae* it decreases by 0.9%. However, for *E. coli* and human, the difference is more significant, with *E. coli* dropping performance by 13.8% and human by 7.3%. Interestingly, the all datasets experiment produced a mid-range WAF value, indicating that it is possible to produce a successful species-agnostic model.

5.3.6 Overview of GP Models

Since GP produces readable models, after evaluating the performance of the proposed methodology, the models generated by GP across different datasets were analyzed. The goal was to identify which are the operators and combinations that GP uses more often, and how they compare across datasets.

The analysis of the models was conducted using the Python library SymPy 1.3 (Meurer *et al.*, 2017) and the Python package Graphviz 0.10.1 (Ellson *et al.*, 2002). SymPy is a library for symbolic mathematics and was used to convert the GP models obtained in each experiment into expressions that are easily parsed. Graphviz is an open source graph visualization software and was used to visualize the obtained GP models.

Table 5.8 summarizes, for the 10 runs performed in each dataset, the average length (number of tree nodes) of the models and the average relative frequency of variables BP, CC and MF in the models. These are calculated after arithmetic simplification of the raw models returned by GP.

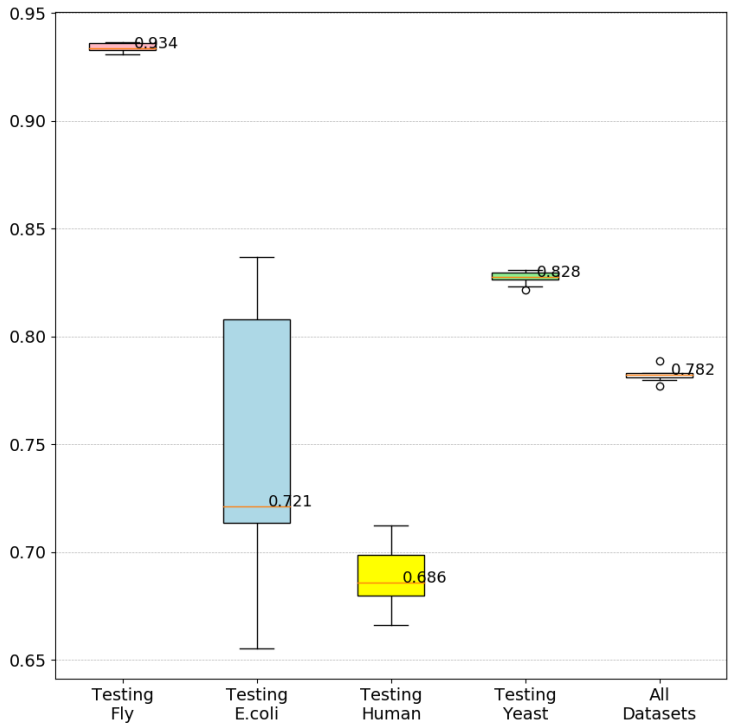

Figure 5.16: WAF boxplot using multispecies data in training set.

Table 5.8: Analysis of GP models for each dataset.

Dataset	BP	CC	MF	Length
STRING-EC	0.281	0.432	0.288	92
STRING-DM	0.477	0.330	0.193	175.6
BIND-SC	0.326	0.473	0.201	185.9
DIP/MIPS-SC	0.437	0.395	0.168	51.6
STRING-SC	0.322	0.533	0.145	70
DIP-HS	0.432	0.372	0.196	137
STRING-HS	0.550	0.268	0.182	72.2
GRID/HPRD-unbal-HS	0.411	0.306	0.283	58.6
GRID/HPRD-bal-HS	0.457	0.296	0.247	51.6
Average	0.410	0.378	0.211	99.8
Species-agnostic	0.508	0.373	0.134	89.9

As expected, variable MF appears less frequently in the GP models, except for the STRING-EC dataset. Once again, the explanation for this exception can be the lack of BP annotations for the species *E. coli*. These results are in agreement with the previous results that indicated

that BP and CC annotations are stronger indicators for PPI than MF annotation. However, the frequency in which a given variable appears in a GP model does not necessarily measure its importance for the predictions, as its effect may be stronger or weaker depending on its surrounding context. The average length of the GP models is 99.8, with somewhat large differences between datasets. One interesting observation is that, when the datasets are smaller, such as STRING-DM and BIND-SC, the average length of the GP models tends to increase. This may be an indication that GP is evolving highly tuned, possibly overfitted models, for lack of sufficient data to induce smaller and more general ones. However, in GP the complexity of a model does not depend on its size, but on the particular features and operators used to build it, and therefore one cannot assume that larger models overfit more than smaller ones (Silva *et al.*, 2012).

In GP models of the species-agnostic experiment the differences between the frequencies of the variables BP, CC and MF are more significant, being MF the least frequent variable and BP, clearly, the most frequent variable (last row of Table 5.8). Once again the results indicate that similarities in BP and CC annotations are stronger indicators for PPI than MF annotation, with a slight advantage for BP.

5.3.7 Comparison with other PPI Prediction Methods

By using benchmark datasets, our results could be in principle directly compared to the results obtained by other works using the same datasets. However, our results cannot be directly compared to the published ones, first because we used more recent versions of the GO KG, and second because we excluded some protein pairs of the benchmark datasets. The results obtained in different works are also not directly comparable between themselves. Nevertheless, the results from relevant related work were compiled, to support a comparative overview.

Table 5.9 summarizes the area under the receiver operating characteristic curve (AUC-ROC) for several prediction methods and the AUC-ROC median for the proposed methodology using the best SSM.

The results in the third to sixth columns are all based on a similar approach, whereby an interacting protein pair is described by a vector that combines the presence/absence of GO terms for both proteins. The ULCA (up to lowest common ancestors) variant takes all annotations, direct and inherited up to the lowest common ancestor. The AA (all ancestors) variant takes all annotations, direct and inherited. The weighted variants (WULCA and WAA) weight the presence of a GO term by its IC. This is not a semantic-similarity based approach, but rather a propositional feature vector approach over the GO KG. The third column shows the best prediction performance of the ULCA with a Naïve Bayes classifier using the BP aspect obtained by Maetschke *et al.* (2011). The fourth, fifth, and sixth columns present the results obtained by cross-validation of SVM obtained by Bandyopadhyay & Mallick (2017) using all aspects.

The seventh column refers to an improved algorithm proposed by Maetschke *et al.* (2011) to compute SS between GO terms annotated to proteins in benchmark interaction datasets. The eighth column reports the results obtained by Chen *et al.* (2019). Here, an ensemble learning approach for PPI prediction integrates five learning algorithms and three types protein-pair features based on the physicochemical properties of amino acids, GO, and interaction network topologies.

Bandyopadhyay & Mallick (2017) is one of the most recent works where the impact of the GO KG updates introduces less bias in a comparison with our results. We can observe that our approach achieves a 5-6% lower performance in the STRING-SC and STRING-EC datasets, but an equal performance in the human dataset, while in the STRING-DM dataset it achieves an 11% higher performance. An important difference between Bandyopadhyay and Mallick's approach and ours, is that while ours uses SS as the features characterizing a protein pair, they employ IC weighted vectors of the GO terms assigned to each protein. Their approach gives the machine learning algorithm access to the annotations themselves, with models being able to learn exactly which annotations are better interaction predictors, while in our approach the model is only able to learn which semantic aspects are the best predictors.

Table 5.9: Comparison of the evolved combinations (EC) with other PPI prediction methods.

Dataset	EC	ULCA by Maetshke *et al.*	ULCA by Bandyopadhyay and Mallick	WULCA by Bandyopadhyay and Mallick	WAA by Bandyopadhyay and Mallick	Best SSM by Jain and Bader	Tool by Chen *et al.*
STRING-SC	0.90	0.83	0.92	0.95	0.95		0.95
STRING-HS	0.95	0.85	0.90	0.93	0.95		0.92
STRING-EC	0.91	0.93	0.93	0.96	0.96		0.95
STRING-DM	0.97	0.82	0.82	0.86	0.85		0.92
DIP-HS	0.97					0.92	
BIND-SC	0.96			0.96	0.94		
DIP/MIPS-SC	0.88			0.93	0.93		
GRID/HPRD-bal-HS	0.74			0.68	0.67		
GRID/HPRD-unbal-HS	0.81			0.83	0.82		

The Onto2Vec method, proposed by Smaili *et al.* (2018), is also applied to predict PPIs in human and yeast. Although they did not use our benchmark datasets, PPIs were collected from STRING, the same database of PPIs from STRING-SC and STRING-HS datasets. In this work, Onto2Vec was used to learn feature vectors for proteins combining information about their GO annotations and the semantics of the GO classes in a single representation. The best AUC-ROC values were 0.8869 and 0.8931 for yeast and human datasets, respectively, and were obtained using an artificial neural network on the Onto2Vec representations.

Chapter 6

Conclusions and Future Work

KGs represent a unique opportunity for machine learning and knowledge discovery, given the growing number of KGs and their ability to provide meaningful context to the data through semantic representations. SS as a KG-based representation has been used to support several biomedical applications, ranging from the prediction of PPIs, of gene product function or even of genes associated with diseases. Using KG-based SSMs typically includes selecting the aspects of the KG that are relevant for a given target application, a task that needs expert knowledge. However, in complex and multi-domain data, such as biomedical data, this manual selection can represent a challenge, since for some learning tasks identifying the most relevant semantic aspects is not straightforward. Adjusting the selection of SS aspects to the machine learning task in an automated fashion, until now, had not yet been tackled.

This chapter summarizes the main contributions of this work and discusses some limitations and future work.

6.1 Summary of Main Contributions

This dissertation focused on overcoming the limitations imposed by manual selection of SS aspects for machine learning. We have developed a novel approach, using GP, that is able to learn suitable combinations of SS aspects to support supervised learning. GP was chosen for its unmatched ability to search large solution spaces by means of evolving a population of free-form models through crossover and mutation. An analysis of the related work showed that there are no known approaches that use GP to improve SS representations.

The developed approach was applied and evaluated in PPI prediction, using the GO as the KG (with its three semantic aspects: BP, CC and MF) and a set of nine benchmark datasets. SS derived from GO is one of the most widely used indicators for protein interaction. Furthermore, the relationships between the different semantic aspects and PPI interaction are well established.

The first step in the implementation of our methodology was the computation of SSM for each GO semantic aspect. This computation was done using six distinct SSMs, providing a comparative evaluation of SSMs for PPI prediction in the nine benchmark datasets. The goal

of this evaluation was to elucidate which SSM is more suitable for PPI prediction. The results showed that the SSMs that use Resnik approaches have better performance in general (section 5.3.1).

In the second and third steps, for each dataset, GP was used over the obtained SS values for each GO semantic aspect to select the best combination. This combination was then used for PPI prediction on test set. The results showed that, for sufficiently large datasets (greater than 2000 protein pairs), GP is able to learn suitable combinations of SS aspects that improve PPI prediction performance over classical static combinations. Thus, a GP-based approach is able to improve over the manual selection of semantic aspects (section 5.3.2). Since results suggested that the lower size of datasets could explain the lower performance of our methodology, we investigated the potential impact of the size of datasets by combining datasets with data from the same species. These experiments yielded very interesting results and confirm that using larger datasets for training improve the performance of classification (section 5.3.3).

GO and GO annotations are designed to be species-independent and this is especially important for the cases where the number of documented proteins of that species is low. Therefore, and since we used benchmark PPI datasets from four different species, we also demonstrated that a model trained on one or multiple other species can be transferred and successfully be applied to a different species (sections 5.3.4 and 5.3.5).

One of the main advantages of GP and one of the reasons that led us to choose this machine learning algorithm is its ability to produce readable models. The analysis of the models generated across the different datasets showed that these models reflect the expert knowledge in PPI domain, proving our hypothesis that the combination of semantic aspects can be selected automatically using GP (section 5.3.6).

The main conclusion of this work is that our methodology can be used to automatically select the combination of SS aspects to the machine learning task. For PPI prediction, when compared with manual selected combinations, our methodology improves the classification performance.

6.2 Limitations

The promising results reported herein should be considered in the light of some limitations.

The primary limitation of the results concerns the data sources. Although data sources like the GOA database contain consolidated knowledge, the information which they provide is intrinsically biased towards well studied proteins and genes. In addition, it is also probable that PPI datasets will have their own biases, for instance towards the preparation of negative examples.

The approaches based on SS also have some possible limitations. The information in the ontology is used to define the similarity but this information is reduced to a single point (the SS value). Thus, the output of SSMs contains no explicit information of the ontology. Furthermore, SSMs are generally designed by an expert based on a set of assumptions about how an ontology is used and what should constitute a similarity. However, different similarity measures perform well on some datasets and tasks and worse on others, without any measure showing clear superiority across multiple tasks, as also noted in Smaili *et al.* (2018).

Finally, as said before, GP, like nature, is a stochastic process and it can never guarantee the optimal result. In addition, there is high variability among the raw models returned in different runs, even when using the same settings and same dataset. This characteristic raises some difficulty in interpreting the ensemble of GP models.

6.3 Future Work

Despite the narrow application proposed in this dissertation, the proposed methodology is not specifically designed for PPI prediction and can be generalized to other applications and domains available as KG. Disease gene discovery and prioritization using the HPO, or link prediction over KGs, are examples of applications. Furthermore, other SSMs can be explored. As stated in Section 2.1.4, there are many SSMs applied to biomedical ontologies that have been proposed but have not been tested.

Another direction for future work is combining our approach for semantic aspect selection with other semantic representations such as graph embeddings and graph kernels. These state-of-art semantic representations are static and take into consideration all semantic aspects, ignoring that some may be irrelevant to the downstream learning task, potentially introducing noise. Thus, GP can be used to learn suitable semantic of entities extracted from KGs capable of supporting a specific supervised learning task. In other words, GP can be used to learn which properties or portions of the graph are more relevant and how to combine them to produce adaptive semantic representations to address a given machine learning task.

References

Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M. *et al.* (2004). UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, **32**, D115–D119. 24

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T. *et al.* (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, **25**, 25–29. 44

Bandyopadhyay, S. & Mallick, K. (2017). A new feature vector based on gene ontology terms for protein-protein interaction prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **14**, 762–770. 16, 49

Belleau, F., Nolin, M.A., Tourigny, N., Rigault, P. & Morissette, J. (2008). Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, **41**, 706–716. 6

Ben-Hur, A. & Noble, W.S. (2005). Kernel methods for predicting protein–protein interactions. *Bioinformatics*, **21**, i38–i46. 25

Ben-Hur, A. & Noble, W.S. (2006). Choosing negative examples for the prediction of protein-protein interactions. *BMC Bioinformatics*, **7**, S2. 40

Berners-Lee, T., Hendler, J., Lassila, O. *et al.* (2001). The semantic Web. *Scientific American*, **284**, 28–37. 5

Bizer, C., Heath, T. & Berners-Lee, T. (2011). Linked data: The story so far. In A. Sheth, ed., *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 205–227, IGI Global, Hershey, PA, USA. 5

Bloom, J.D. & Adami, C. (2003). Apparent dependence of protein evolutionary rate on number of interactions is linked to biases in protein–protein interactions data sets. *BMC Evolutionary Biology*, **3**, 21. 44

Chen, K.H., Wang, T.F. & Hu, Y.J. (2019). Protein-protein interaction prediction using a hybrid feature representation and a stacked generalization scheme. *BMC Bioinformatics*, **20**, 308. 16, 49

REFERENCES

Côté, R.G., Jones, P., Martens, L., Kerrien, S., Reisinger, F., Lin, Q., Leinonen, R., Apweiler, R. & Hermjakob, H. (2007). The protein identifier cross-referencing (PICR) service: reconciling protein identifiers across multiple source databases. *BMC Bioinformatics*, **8**, 401. 25

Couto, F.M., Silva, M.J., Lee, V., Dimmer, E., Camon, E., Apweiler, R., Kirsch, H. & Rebholz-Schuhmann, D. (2006). GOAnnotator: linking protein GO annotations to evidence text. *Journal of Biomedical Discovery and Collaboration*, **1**, 19. 15

De Raedt, L. (2008). *Logical and relational learning*. Springer-Verlag, Berlin Heidelberg, Germany. 1

De Vries, G.K. (2013). A fast approximation of the Weisfeiler-Lehman graph kernel for RDF data. In H. Blockeel, K. Kersting, S. Nijssen & F. Železný, eds., *Machine Learning and Knowledge Discovery in Databases*, 606–621, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany. 13

De Vries, G.K.D. & De Rooij, S. (2013). A fast and simple graph kernel for RDF. In *Proceedings of the 2013 International Conference on Data Mining on Linked Data - Volume 1082*, DMoLD'13, 23–34, CEUR-WS.org, Aachen, Germany. 13

De Vries, G.K.D. & de Rooij, S. (2015). Substructure counting graph kernels for machine learning from RDF data. *Journal of Web Semantics*, **35**, 71–84. 14

Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M. & Ashburner, M. (2007). ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, **36**, D344–D350. 7

Domingos, P. (2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books, Inc., New York, NY, USA. 10

Duan, Z.H., Hughes, B., Reichel, L., Perez, D.M. & Shi, T. (2006). The relationship between protein sequences and their gene ontology functions. *BMC Bioinformatics*, **7**, S11. 15

Ehrlinger, L. & Wöss, W. (2016). Towards a definition of knowledge graphs. In M. Martin, M. Cuquet & E. Folmer, eds., *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016, Leipzig, Germany, September 12-15*, vol. 1695 of *CEUR Workshop Proceedings*, CEUR-WS.org. 2, 7

Ellson, J., Gansner, E., Koutsofios, L., North, S.C. & Woodhull, G. (2002). Graphviz – open source graph drawing tools. In P. Mutzel, M. Jünger & S. Leipert, eds., *Graph Drawing*, 483–484, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany. 47

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, **17**, 37. 1

Freudenberg, J. & Propping, P. (2002). A similarity-based method for genome-wide prediction of disease-relevant human genes. *Bioinformatics*, **18**, S110–S115. 15

Guzzi, P.H., Mina, M., Guerra, C. & Cannataro, M. (2011). Semantic similarity analysis of protein data: assessment with biological features and issues. *Briefings in Bioinformatics*, **13**, 569–585. 10

Harispe, S., Ranwez, S., Janaqi, S. & Montmain, J. (2013). The semantic measures library and toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies. *Bioinformatics*, **30**, 740–742. 20

Harispe, S., Sánchez, D., Ranwez, S., Janaqi, S. & Montmain, J. (2014). A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain. *Journal of Biomedical Informatics*, **48**, 38–53. 10

Hoehndorf, R., Schofield, P.N. & Gkoutos, G.V. (2011). PhenomeNET: a whole-phenome approach to disease gene discovery. *Nucleic Acids Research*, **39**, e119–e119. 15

Huntley, R.P., Sawford, T., Mutowo-Meullenet, P., Shypitsyna, A., Bonilla, C., Martin, M.J. & O'donovan, C. (2014). The GOA database: gene ontology annotation updates for 2015. *Nucleic Acids Research*, **43**, D1057–D1063. 24

Jain, S. & Bader, G.D. (2010). An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. *BMC Bioinformatics*, **11**, 562. 15, 25

Kiryakov, A., Popov, B., Terziev, I., Manov, D. & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, **2**, 49–79. 6

Köhler, S., Schulz, M.H., Krawitz, P., Bauer, S., Dölken, S., Ott, C.E., Mundlos, C., Horn, D., Mundlos, S. & Robinson, P.N. (2009). Clinical diagnostics in human genetics with semantic similarity searches in ontologies. *The American Journal of Human Genetics*, **85**, 457–464. 15

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA. 28

Koza, J.R. (2010). Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines*, **11**, 251–284. 10

Lee, P.H. & Lee, D. (2005). Modularized learning of genetic interaction networks from biological annotations and mRNA expression data. *Bioinformatics*, **21**, 2739–2747. 15

Lei, Z. & Dai, Y. (2006). Assessing protein similarity with gene ontology and its use in subnuclear localization prediction. *BMC Bioinformatics*, **7**, 491. 15

Li, M., Li, Q., Ganegoda, G.U., Wang, J., Wu, F. & Pan, Y. (2014). Prioritization of orphan disease-causing genes using topological feature and GO similarity between proteins in interaction networks. *Science China Life Sciences*, **57**, 1064–1071. 15

Lin, N., Wu, B., Jansen, R., Gerstein, M. & Zhao, H. (2004). Information assessment on predicting protein-protein interactions. *BMC Bioinformatics*, **5**, 154. 15

Liu, W., Liu, J. & Rajapakse, J.C. (2018). Gene ontology enrichment improves performances of functional similarity of genes. *Scientific Reports*, **8**, 12100. 15

Lösch, U., Bloehdorn, S. & Rettinger, A. (2012). Graph kernels for RDF data. In E. Simperl, P. Cimiano, A. Polleres, O. Corcho & V. Presutti, eds., *The Semantic Web: Research and Applications*, 134–148, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany. 13

Maetschke, S.R., Simonsen, M., Davis, M.J. & Ragan, M.A. (2011). Gene ontology-driven inference of protein–protein interactions using inducers. *Bioinformatics*, **28**, 69–75. 15, 16, 25, 45, 49

Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J.K., Singh, S., Rathnayake, T., Vig, S., Granger, B.E., Muller, R.P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M.J., Terrel, A.R., Roučka, v., Saboo, A., Fernando, I., Kulal, S., Cimrman, R. & Scopatz, A. (2017). SymPy: symbolic computing in Python. *PeerJ Computer Science*, **3**, e103. 47

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, 3111–3119, Curran Associates Inc., USA. 14

Park, Y. (2009). Critical assessment of sequence-based protein-protein interaction prediction methods that do not require homologous protein sequences. *BMC Bioinformatics*, **10**, 419. 24, 45

Patil, A. & Nakamura, H. (2005). Filtering high-throughput protein-protein interaction data using a combination of genomic features. *BMC Bioinformatics*, **6**, 100. 15

Paulheim, H. & Fümkranz, J. (2012). Unsupervised generation of data mining features from linked open data. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, 31:1–31:12, ACM, New York, NY, USA. 16

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830. 20

Perez-Iratxeta, C., Bork, P. & Andrade, M.A. (2002). Association of genes to genetically inherited diseases using data mining. *Nature Genetics*, **31**, 316. 15

Pesquita, C. (2017). Semantic similarity in the gene ontology. In C. Dessimoz & N. Škunca, eds., *The Gene Ontology Handbook*, 161–173, Humana Press, New York, NY, USA. 10

Pesquita, C., Faria, D., Bastos, H., Falcao, A. & Couto, F. (2007). Evaluating GO-based semantic similarity measures. In *Proceedings of the 10th Annual Bio-Ontologies Meeting*, 37–40, Vienna, Austria. 27

Pesquita, C., Faria, D., Falcao, A.O., Lord, P. & Couto, F.M. (2009). Semantic similarity in biomedical ontologies. *PLOS Computational Biology*, **5**, e1000443. 3, 8, 9

Poli, R., Langdon, W.B., McPhee, N.F. & Koza, J.R. (2008). *A field guide to genetic programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`. 3, 10

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, 448–453, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 27

Ristoski, P. & Paulheim, H. (2016a). Rdf2Vec: RDF graph embeddings for data mining. In P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch, F. Lecue, F. Flöck & Y. Gil, eds., *The Semantic Web – ISWC 2016*, 498–514, Springer International Publishing, Cham, Switzerland. 14

Ristoski, P. & Paulheim, H. (2016b). Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Journal of Web Semantics*, **36**, 1–22. 13, 14

Ristoski, P., Bizer, C. & Paulheim, H. (2015). Mining the web of linked data with rapidminer. *Journal of Web Semantics*, **35**, 142–151. 16

Robinson, P.N., Köhler, S., Bauer, S., Seelow, D., Horn, D. & Mundlos, S. (2008). The human phenotype ontology: a tool for annotating and analyzing human hereditary disease. *The American Journal of Human Genetics*, **83**, 610–615. 7, 15

Schmachtenberg, M., Bizer, C. & Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz & C. Goble, eds., *The Semantic Web – ISWC 2014*, 245–260, Springer International Publishing, Cham, Switzerland. 2, 6

Schuurman, N. & Leszczynski, A. (2008). Ontologies for bioinformatics. *Bioinformatics and Biology Insights*, **2**, BBI–S451. 6

Seco, N., Veale, T. & Hayes, J. (2004). An intrinsic information content metric for semantic similarity in WordNet. In *Proceedings of the 16th European Conference on Artificial Intelligence*, ECAI'04, 1089–1090, IOS Press, Amsterdam, The Netherlands. 26

Shervashidze, N. & Borgwardt, K. (2010). Fast subtree kernels on graphs. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams & A. Culotta, eds., *Advances in Neural Information Processing Systems 22*, 1660–1668, Curran, Vancouver, BC, Canada. 14

Silva, S., Dignum, S. & Vanneschi, L. (2012). Operator equalisation for bloat free genetic programming and a survey of bloat control methods. *Genetic Programming and Evolvable Machines*, **13**, 197–238. 49

Sipper, M., Fu, W., Ahuja, K. & Moore, J.H. (2018). Investigating the parameter space of evolutionary algorithms. *BioData Mining*, **11**. 29

Smaili, F.Z., Gao, X. & Hoehndorf, R. (2018). Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations. *Bioinformatics*, **34**, i52–i60. 14, 50, 52

Turner, F.S., Clutterbuck, D.R. & Semple, C.A. (2003). POCUS: mining genomic sequence annotation to predict disease genes. *Genome Biology*, **4**, R75. 15

Tzanis, G., Berberidis, C. & Vlahavas, I. (2008). Biological data mining. In J. Wang, ed., *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*, 1696–1705, IGI Global, Hershey, PA, USA. 1

Whetzel, P.L., Noy, N.F., Shah, N.H., Alexander, P.R., Nyulas, C., Tudorache, T. & Musen, M.A. (2011). BioPortal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic Acids Research*, **39**, W541–W545. 7

Wu, X., Zhu, L., Guo, J., Zhang, D.Y. & Lin, K. (2006). Prediction of yeast protein–protein interaction network: insights from the gene ontology and annotations. *Nucleic Acids Research*, **34**, 2137–2150. 15

Yu, J., Guo, M., Needham, C.J., Huang, Y., Cai, L. & Westhead, D.R. (2010). Simple sequence-based kernels do not predict protein–protein interactions. *Bioinformatics*, **26**, 2610–2614. 25, 42

Zhang, P., Zhang, J., Sheng, H., Russo, J.J., Osborne, B. & Buetow, K. (2006). Gene functional similarity search tool (GFSST). *BMC Bioinformatics*, **7**, 135. 15

Zhang, S.B. & Tang, Q.R. (2016). Protein–protein interaction inference based on semantic similarity of gene ontology terms. *Journal of Theoretical Biology*, **401**, 30–37. 15